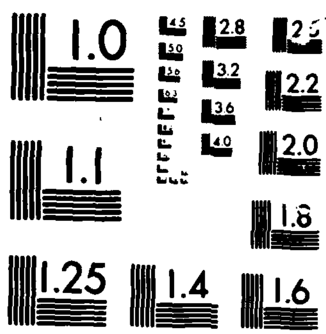
AD-A167 152 COMPUTER MODELING OF THIN FILM GROWTH(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING J A STEFONECK DEC 84 AFIT/GEP/PH/84D-10 1/2

UNCLASSIFIED F/G 7/4 NL



MICROCOP CHART

AD-A167 152

COMPUTER MODELING OF THIN FILM GROWTH

THESIS

Jeffrey A. Steffeneck
Captain, USAF

AFIT/GEP/PH/84D-10

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC
ELECTE
MAY 1 2 1986
S D
B

DTIC FILE COPY

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 5 12 030

AFIT/GEP/PH/84D-10

COMPUTER MODELING OF THIN FILM GROWTH

THESIS

Jeffrey A. Stefoneck
Captain, USAF

AFIT/GEP/PH/84D-10

DTIC
S ELECTE D
MAY 1 2 1986
B

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AFIT/GEP/PH/84D-10

COMPUTER MODELING OF THIN FILM GROWTH

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Engineering Physics

Jeffrey A. Stefoneck, B.S.

Captain, USAF

December 1984

Approved for public release; distribution unlimited

Preface

This study represents an initial effort to give the Air Force Institute of Technology (AFIT) a computer model which simulates thin film growth by vapor-deposition. It was my goal in this effort to establish an initial building block which follow on studies could use to expand and modify. It is my hope that with this documentation and programming, modification and expansion will be a relatively easy task.

Finally, a word of thanks to all of the AFIT Faculty who provided guidance in this effort, especially Major Wharton. Also, I wish to thank my wife Debra for her understanding, concern, and typing, and my sons Brian and Aaron for the quiet hours they gave.

Jeffrey A. Stefoneck

Accession
NTIS GRA
DTIC TAB
Dist
A-1 23

Table of Contents

List of Figures

Abstract

A two dimensional, hard disk computer model ~~has been~~ made which simulates thin film growth. The model represents deposition molecules by hard disks, which are trajected at some angle to the substrate. At the substrate, the model assumes a limited mobility where incident molecules are captured upon contact and then allowed to move to the nearest rest pocket. The model monitors disk movement by organizing the deposition field into a 320 by 240 array.

An analysis of nine different deposition angles shows that structural anisotropy and voids are a natural occurrence of the deposition process. The amount of unfilled space and the anisotropy can be linked to the deposition angle and mobility of the incident particles.

COMPUTER SIMULATION OF VAPOR DEPOSITED THIN FILM GROWTH

I. Introduction

Vapor-deposited thin films play important roles in many technologies, most notably in optics and microelectronics. In optics, thin films are commonly used for anti-reflection coatings and in broad and narrow band-pass filters. The application of thin film optical devices are manifold, as are their structures, which extend from the simplest single coatings to intricate arrangements of 100 or more layers. In microelectronics, thin films also play an important role. Integrated circuit electronics technology, for example, relies heavily upon vapor-deposited thin films for the formation of metal interconnections.

One technique used for producing thin films is vapor-deposition. In this method, a small quantity of material is placed inside a vacuum chamber where it is heated and the surrounding pressure is lowered to a millionth of atmospheric pressure or less. In addition, the vacuum chamber contains a substrate where the temperature is held equal to or less than half the melting point temperature of the material to be deposited. Thus, the material is evaporated and condenses onto the substrate where a thin film is produced.

Crystalline and amorphous thin films produced by this kind of deposition commonly exhibit excess volume. This

volume, in the form of lattice vacancies, pores, and voids is a nearly universal feature of vapor-deposited thin films. The microstructure formed is columnar as observed by microfractography [Ref 5-9, 12, 18, 19]; by transmission electron microscopy [Ref 4, 9, 14-16, 21, 24, 25]; and by small angle electron [Ref 9, 16, 24, 25] and x-ray scattering [Ref 2, 13, 23], and at large scattering angles in amorphous films [Ref 21, 23]. In addition, the formation of the columnar microstructure has been found to depend upon deposition conditions -- substrate temperature, deposition rate, angle of incidence, and vacuum ambient -- as well as upon the material.

The physical properties of vapor-deposited thin films differ greatly from those of bulk material. Magnetic, optical, electrical, and mechanical properties are known to be influenced by the presence of the columnar structure [Ref 10]. In addition, the columnar microstructure, and thereby the physical properties, can be varied through the deposition conditions. Due to this variability in physical properties, thin films are playing an ever increasing role in optics and microelectronics.

Purpose of Study

The purpose of this ~~study~~ was to make a computer simulation of the vapor-deposition process in such a way as to give rise to an amorphous array of molecules with demonstrable anisotropy and void formations.

Assumptions

In this study, various assumptions were made. The most general are presented here. A more complete list and discussion of their reasonability are presented in Chapter II.

1. It is assumed that the physical thin film deposition process can adequately be represented by a two-dimensional model.
2. Each particle (atom, molecule, or collection of molecules) is assumed to travel on a straight line and at an angle A from the substrate normal until it comes in contact with one of the already deposited particles or substrate.
3. It is assumed that the temperature of the substrate or film and the energy of the incoming particles are such that once they contact a particle (substrate or film), they will stick. Furthermore, the incident particle is assumed always to remain in contact with the particle with which it first made contact. However, the incident particle is allowed to relax to the extent that it moves about the perimeter of the contacted particle until it makes contact with the next closest particle (substrate or film).

General Approach

This study was conducted in eight separate steps: model definition, assumption identification, scope definition,

mathematical model development, program development, program testing, simulation activation, and results analysis. In general, the order of implementation was the same. However, as in most studies, some problems encountered required taking a step or two backwards and reaccomplishing them.

Summary of Current Knowledge

Several properties of thin film columnar growth have been observed and can be used to verify theoretical results. First, for oblique incidence deposition, the columns are found to be inclined toward the vapor source and form an angle B with the substrate normal (See Figure 1.1). The angle formed by the vapor beam and substrate normal is A and is related to B by

$$2 \tan B = \tan A,$$

which has become known as the Tangent Rule [Ref 17].

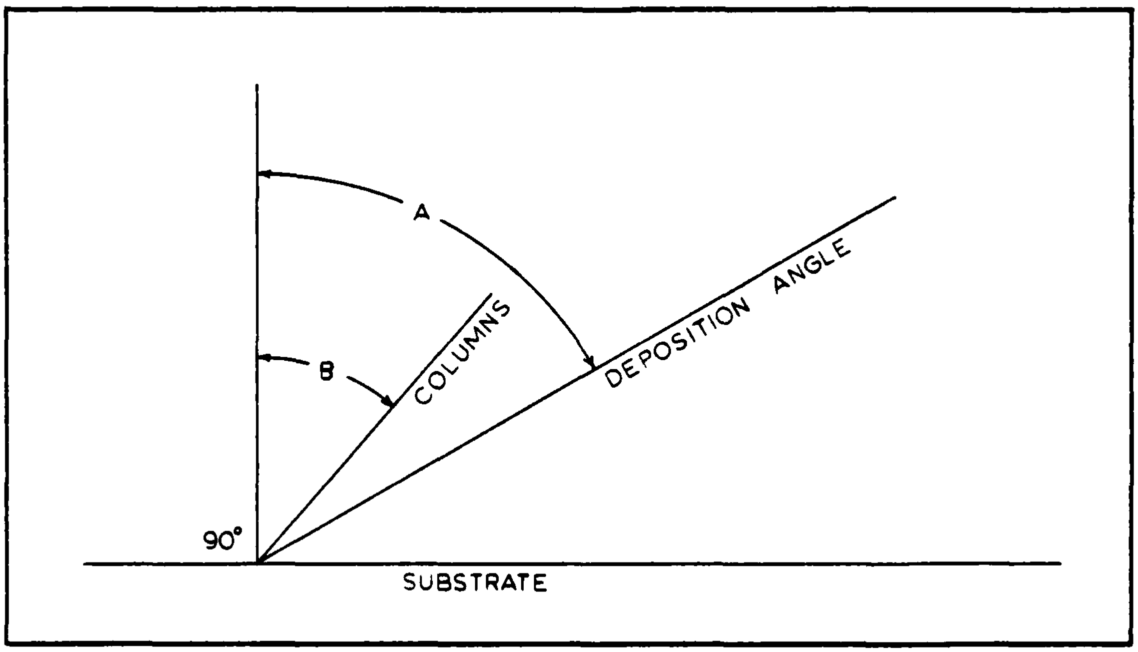


Fig. 1.1 Angles of Tangent Rule

Second, oblique incidence deposition is often accompanied by a change in column shape. While the columnar cross section of normally deposited films are invariably equiaxed, oblique deposition yields columns that are elongated in the direction perpendicular to the plane of incidence (plane determined by the vapor beam and foil normal). Third, although more prominent in obliquely deposited films, the columnar structure persists under conditions of normal incidence deposition. Finally, an increase in A leads to an increase in the spacing between columns and a decrease in the frequency of column branching.

Sequence of Presentation

The results of this study will be presented in the five remaining chapters and two appendixes. In Chapter II, current literature will be reviewed and a physical model developed. Then some simplifying assumptions will be made and the resulting computer model presented. Chapter III will present the computer program and how the model is implemented, while Chapter IV will discuss the computer program validation. Chapter V will show the results of some simulation runs along with their analysis. All conclusions and recommendations will be presented in Chapter VI. The appendixes contain two types of information which are of great importance to follow on studies but is of little use to the casual reader. Appendix A contains the complete Fortran listing of the computer program with all of the built-in self testing procedures.

Appendix B contains various derivations. Since most derivations concerning this simulation are common algebra derivations, they were put in the appendixes for those people that wish to be reacquainted with them.

II Analysis

The purpose of this study was to make a computer simulation of the vapor-deposition process in such a way as to give rise to an amorphous array of molecules with demonstrable anisotropy and void formations. To accomplish this goal, an adequate model of the physical process needed to be constructed. This was carried out in three phases. The first was a literature search for any applicable experimental results or observations. Then the data collected was analyzed and some assumptions were made. Finally, the model was constructed. The applicable portions of these phases are presented in capsulized form in the remainder of this chapter.

Applicable Experimental Results and Observations

Crystalline and amorphous thin films produced by vapor deposition commonly exhibit a columnar microstructure and excess volume. The volume, in the form of lattice vacancies, pores, and voids is a nearly universal feature. The microstructure formed is columnar as observed by microfractography [Ref 5-9, 12, 18, 19]; by transmission electron microscopy [Ref 4, 9, 14-16, 21, 24, 25]; and by small angle electron [Ref 9, 16, 24, 25] and x-ray scattering [Ref 2, 13, 23], and at large scattering angles in amorphous films [Ref 21, 23]. One example of the columnar microstructure is that revealed by Nieuwenhuizen and Haanstra [Ref 17] by microfractography. Figure 2.1 shows one of their fractographs of the structure

in an aluminum film that was produced by oblique incidence deposition in a 10^{-5} Torr atmosphere.

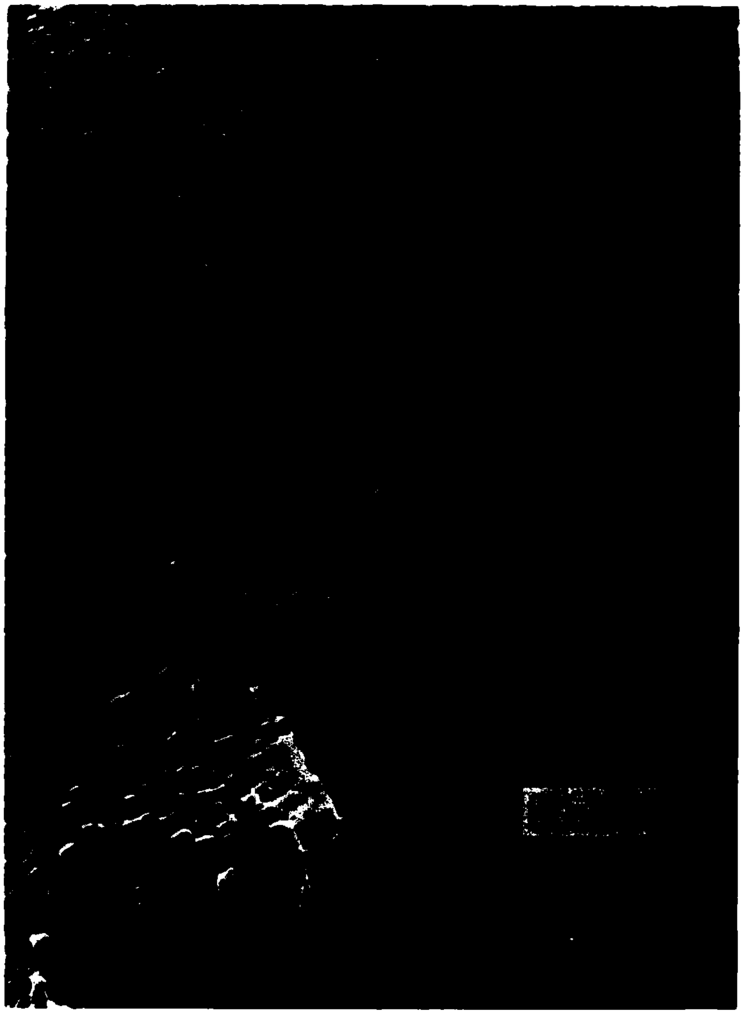


Fig. 2.1. Microfractograph [Ref 17] of Fractured Edge of 1μm thick Al Film

The angle at which these columns grow has also been studied. Nieuwenhuizen and Haanstra [Ref 17] first reported careful determinations of this over a wide range of deposi-

tion angles and found that their results could be well described by what has become known as the "tangent rule".

$$2 \tan B = \tan A$$

Here B is the angle formed between the column growth and the substrate normal, and A is the angle formed between the source direction and the substrate normal (See Figure 2.2a). Figure 2.2b shows the columnar growth angle versus deposition angle as a result of the tangent rule, and as a result of a columnar growth angle equal to the deposition angle (a common misconception).

Although the tangent rule was obtained by experimental observation, much data has been accumulated to support it in general. Leamy et al [Ref 10:317] has compiled data from many sources on depositions of crystalline and amorphous thin films, and has found that the tangent rule was obeyed by all data considered at $0^\circ \leq A \leq 60^\circ$. In fact, only the results of Nakhodkin and Shaldervan at $A > 60^\circ$ does not fall within acceptable ranges of the tangent rule. However, in recent years some experimentalists have found that the tangent rule may not apply under certain conditions [Ref 10].

In conjunction with studies on columnar growth angle, Nakhodkin and Shaldervan [Ref 15:22-24] have noted that the formation of columns depend upon the deposition conditions (substrate temperature, deposition rate, angle of incidence and vacuum ambient) as well as upon the material itself. Leamy et al [Ref 10:312] have noted from source considered, that columnar structures were observed only when the mobility

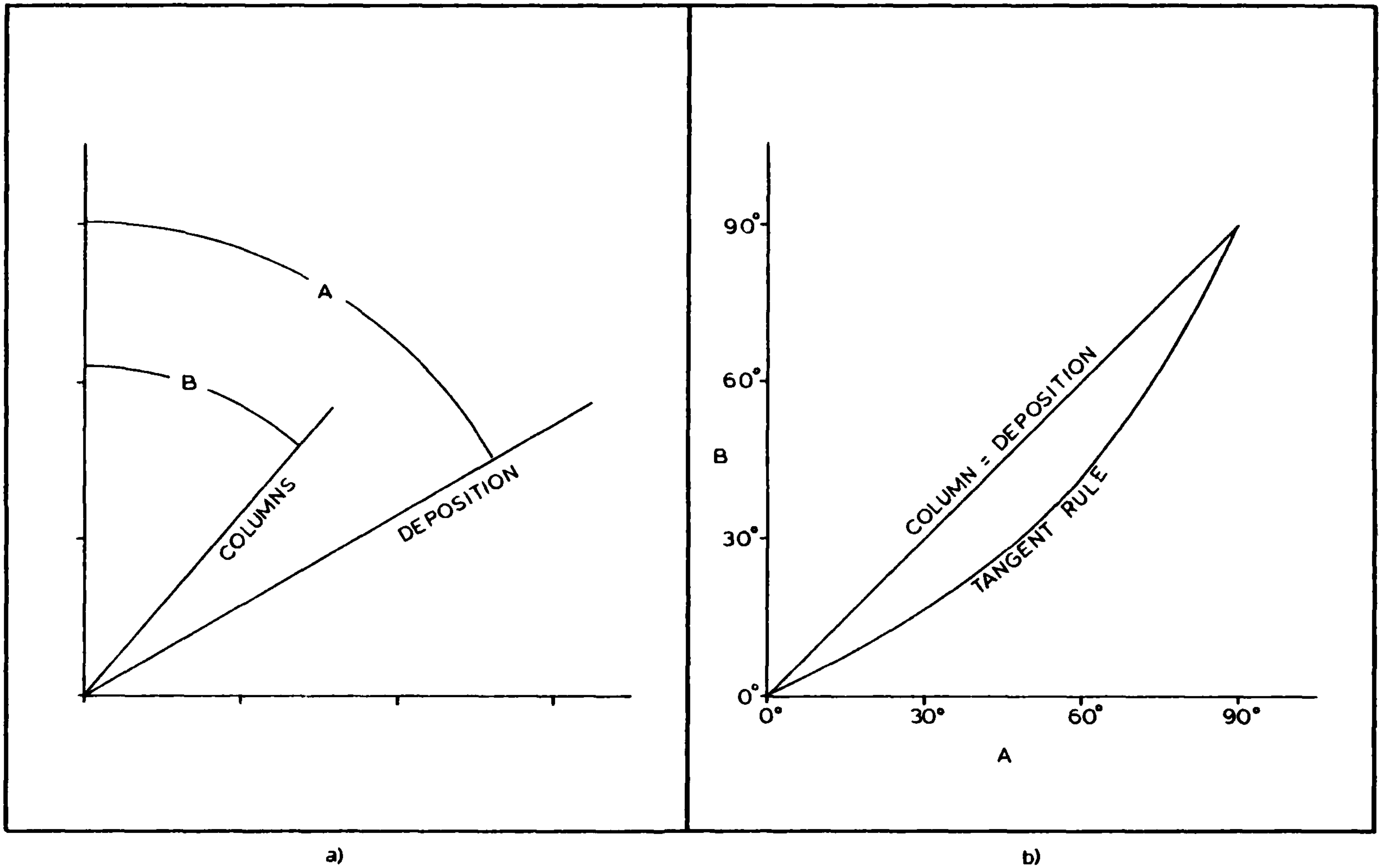


Fig. 2.2. Tangent Rule

(capability of the particles to move once attached to the substrate or film) of the deposited atoms were "limited". For example, columns are observed in films of high melting point materials (chromium, beryllium, silicon and germanium), in compound materials of high binding energy (CdTe, CAF_2, and PbS), and in non-noble metals evaporated in the presence of oxygen (aluminum, iron, and N_1-Fe). Amorphous films (silicon, germanium, S_1O, and RE-TM alloys), whose existence in the amorphous state depends upon a limited atomic mobility, universally exhibit a columnar structure when deposited at sufficiently low temperatures. It has also been noted that depositions with successively increasing substrate temperatures eventually lead to the elimination of the columnar structure [Ref 10:317].

In addition to the angle of the columnar microstructure and a limited mobility, other characteristics have also been found. For example, oblique incidence deposition is often accompanied by a change in column shape. While the columnar cross section of normally deposited films are invariably equiaxed, oblique deposition yields columns that are elongated in the direction perpendicular to the plane of incidence (plane determined by the vapor beam and the foil normal) [Ref 10:13]. As angle of incidence increases, the column shape becomes more elliptical and the void regions between columns become thinner along a direction parallel to the incidence direction. This increase in void network size is reflected in density measurements, which show a monotonic

decrease in density with increasing A [Ref 15:24, 21]. Finally, although more prominent in obliquely deposited films, the columnar structure persists under conditions of normal incidence deposition [Ref 9, 15, 16].

<u>Mobility and Shadowing</u>

In vapor deposition, direct condensation of vapor atoms occurs on substrates at temperatures equal to or less than half the melting point temperature of the vapor species. This causes vapor-solid reactions to proceed under highly nonequilibrium conditions. Consequently, the rate for evaporation is insignificant relative to the rate of condensation. Therefore, for the sake of simplicity, it could be assumed without unreasonable error that incoming particles are captured by the substrate at first impact.

After the particle is captured by the substrate, whether on the first or whatever encounter, another question needs to be answered. How much can the particle move on the substrate before settling? Does it have an infinite, low, or no mobility? Consider first the case of infinite substrate mobility. In this case, a uniform type of material would be formed. Similar to marbles in a box, a dense pack configuration results. A crude example of this would be the formation of solids by the freezing of liquids. The last case, of no substrate mobility, produces results just as unacceptable. Here the columnar structure is formed, but the chainlike structure formed would produce a material density which would

be unrealistic. This leaves only the case of low or limited mobility as an acceptable possibility. Also this bears out Nakhodkin and Heinemann's observation that columnar structures are possible only when the mobility of deposited atoms were limited.

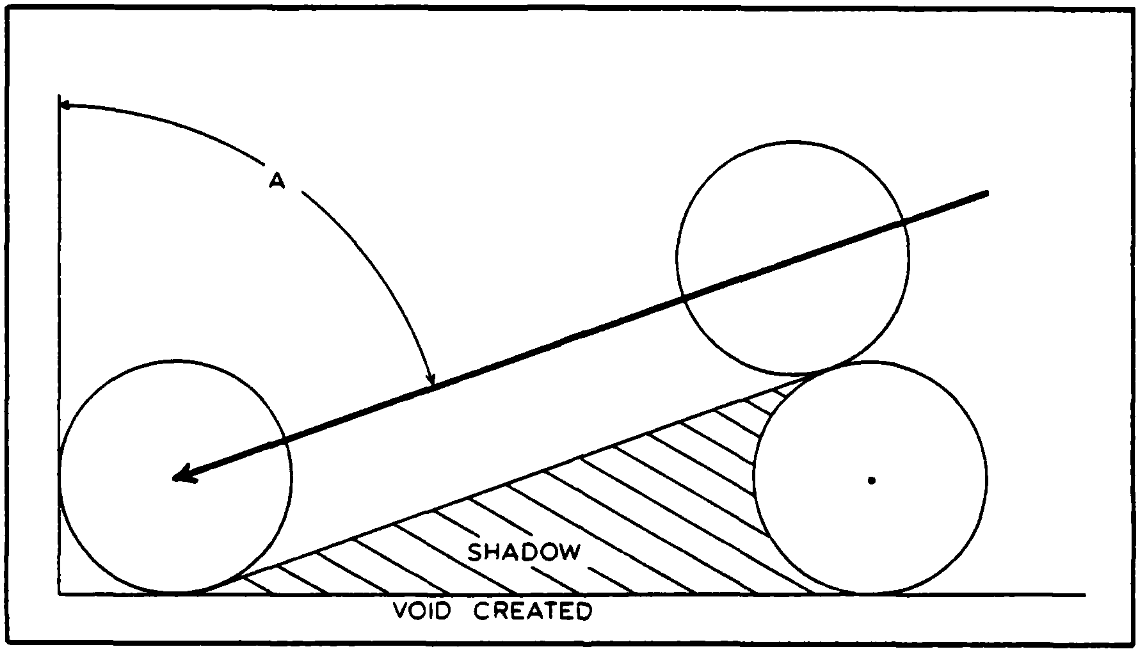


Fig. 2.3. Particle Shadowing

If limited mobility is assumed, the formation of voids can be understood with little difficulty. Consider the finite size particles shown in Figure 2.3. Particle x represents a substrate protrusion. If particle y was trajected to the left of particle x with a trajectory angle of A, the closest that it could impact would be some distance away. In other words, the substrate particles that are directly exposed to the vapor beam shield or shadow unoccupied sites. Thus voids are created. If migration of the particles after condensation does not fill up the voids (limited mobility),

the void structure is maintained and grows with the subsequent deposition of particles.

The consequences of shadowing and limited mobility also produce other effects which have been noted in experimental results. One case in point is the decrease in density experienced with increasing A [Ref 15:24, 21]. It is easy to see from Figure 2.3, that an increase in A would also increase the void width and therefore would decrease the density.

A Model of the Thin Film Growth Process

Thin film growth by vapor-deposition can be pictured quite accurately in light of the previous section. The process is described here in the sequential order that a particle being deposited would see it.

In the vapor-deposition process, particles (atoms, molecules, or collection of molecules) are ejected at a point above the substrate with some determined rate and angle A from the substrate normal. Each particle moves thermodynamically. As a whole, the particles comprise a vapor beam which moves at angle A toward the substrate. After traveling for some time, particles of the vapor beam encounter the substrate. Whether they attach themselves, bounce, attach and release again, etc. is determined by the energies of the incoming particle and the substrate particles contacted. Once the particle is attached to the substrate, the particle is permitted to move on the substrate. However, the movement of the particle on the substrate is limited. How limited

again depends on the energy that the particle still possesses and the energies of the substrate particles on which it encounters. Eventually, the particle becomes affixed to the substrate. As more and more particles attach themselves to the substrate and settle, the film grows.

Computer Model

In an attempt to build a starting block computer model, many assumptions need to be made and justified. Once this is done a more simplified model can be constructed. In this section, all assumptions will be presented along with a short justification and the ensuing computer model described.

1. It is assumed that the thin film deposition process can adequately be represented by a two dimensional model. -- *This assumption is reasonable since most* quantities desired happen in a plane determined by the substrate normal and the vapor beam. Shape of the columnar growth is the exception.
2. The incident particles are assumed to be disks of a constant diameter D. -- Experimental results of many different substances yield similar microstructures, and therefore suggest that shape does not play an important role in thin film growth.
3. The rate at which particles are ejected from the vapor beam source is small enough that they can be considered as serial events. -- If the mean free path and arrival rate of the ejected particles are

calculated as described in Reference 1, it is easy to see that interaction of incoming particles is rare. Thus the particles can be considered one at a time without any real problem with particle interaction.

4. Each particle is assumed to travel on a straight line and at an angle A from the substrate normal until it comes in contact with the substrate or one of the already deposited particles. -- Again, if the mean free path of the vapor particles is calculated as described in Reference 1, it is easy to see that interaction is rare and travel should be in a straight line. In addition, if the impact points are chosen randomly in the model, this should introduce little error.

5. It is assumed that the temperature of the substrate or film and the energy of the incoming particles are such that once they contact a particle (substrate or film), they will stick. Furthermore, limited mobility is assumed; the incident particle always remains in contact with the particle with which it first made contact, however, it is allowed to relax by moving around the perimeter of the contacted particle until it makes contact with the next closest particle (substrate or film). -- The reasonability of this assumption was discussed previously in the section on mobility and shadowing.

Simplified Computer Model

In the computer model of the vapor deposition process, disks are trajected serially in a straight line to the substrate at angle A from the substrate normal. The x axis start point of each disk trajectory is randomly selected. When an incoming disk makes contact with a substrate disk they stick. The incoming disk then moves around the perimeter of the contact disk until it contacts the next closest disk (substrate or film). Finally, the movement of that incoming disk becomes restricted after two contacts are made, and the process starts over again with another disk being trajected.

III Thin Film Growth (TFG) Simulator

A computer simulation of TFG requires the ability to perform at least two tasks. Obviously one task is to simulate the deposition process. The second is a prerequisite required for the deposition process to occur. It is the task of providing a substrate (a collection of particles on which incident molecules or atoms come to rest). If a substrate is provided and the deposition process can be simulated, the basic ingredients for a TFG simulator are at hand.

In addition to the basic requirements for TFG simulation, analysis programs or methods must also be available if a TFG simulation is to be useful. Exactly what the analysis programs or methods consist of are determined by the information sought. In this particular case, the structure of the thin film is of interest. More specifically, this analysis consists of getting the density, the angle of columnar growth, and a graphic representation of the thin film. The simulator must then contain the ability to perform the basic requirement of TFG simulation and the required analysis.

The TFG simulator, as presented in Appendix A, has the ability to do four of the five tasks mentioned above. It can provide substrates, simulate deposition, calculate density, and solve for the angle of columnar growth. Since software is available that will give a graphic representation of particle locations, this feature was not included. However, the TFG simulator presented in Appendix A together with S (a

statistical and graphics system from Bell Laboratories) satisfies all the requirements established.

The TFG simulator program is composed of a main program interface loop and four subprograms; the buffer editor, the matrix manager, the TFG depositor, and the analyzer. Functionally, the TFG simulator has only three parts: the substrate builder, the TFG depositor, and TFG analyzer. This structure and program control flow is shown in Figure 3.1.

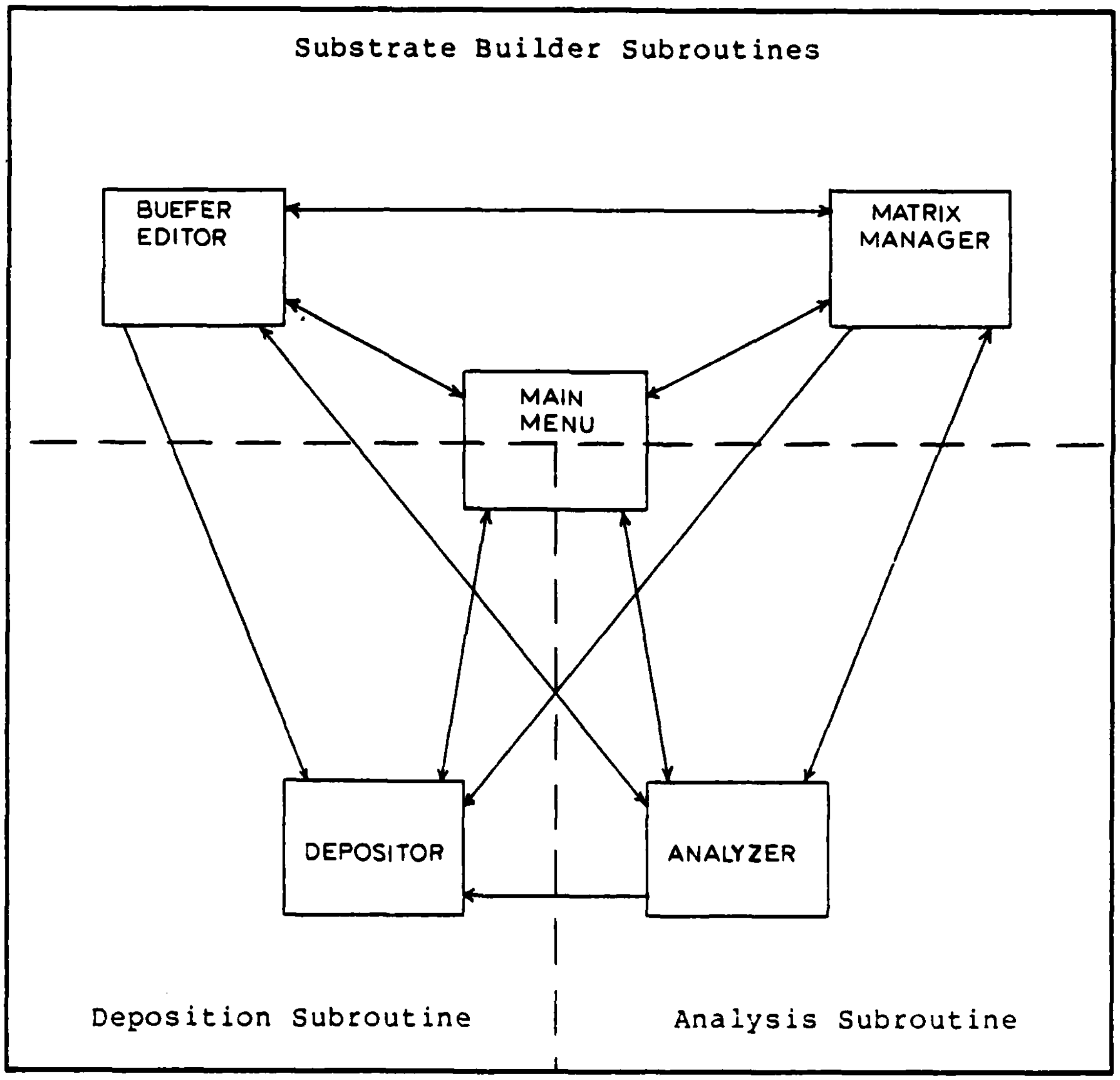


Fig. 3.1. TFG Structure and Program Control Flow

Substrate Handler

The substrate handler provides the TFG simulator the ability to create, modify, store, and recall substrates. It is formed from two TFG simulator subprograms; the buffer editor and the matrix manager. The ability for it to store or recall substrates comes from the matrix manager, whereas the ability for it to create or modify substrates arises through the combined effort of the buffer editor and the matrix manager.

The buffer editor is a convenience interface subprogram. It allows for direct human interface by providing a 240 line buffer which can be filled with substrate building commands. This thereby relieves the user of entering the location of every particle in the substrate. For example, a flat substrate may consist of several hundred particles. The buffer editor provides a means of describing the location of all these particles with just two short line commands of usually less than 25 keystrokes.

The matrix manager provides user control of matrix size and data transfers in or out of the simulator. This is accomplished by several user selectable matrix manager subprograms. The set axes subprogram can set the xy field matrix to 320x240, 160x120, or 80x60 unit cells. Data transfers are accomplished by the read/write subprograms which read and write external field and substrate files. In addition, matrix manager subprograms do all calculations necessary to convert buffer commands to disk locations, which can

then be added to, deleted from, or be entered for the first time into the xy field matrix.

The substrate handler theory of operation is quite simple. It assumes that any given two-dimensional substrate can be represented by a series of single disks, line of disks, or any combination thereof. Examples of these are shown in Figure 3.2. Single disks are entered into the buffer by entering its x and y coordinates. A line of N disks is entered as a length of N disks and a standard angle from the positive x axis as shown in Figure 3.3. The starting point of each line is the last point or the last point of the last line entered. For example, the line in Figure 3.3 could be entered through the buffer by first specifying disk 1 and then specifying the remaining segment. These two buffer commands are listed in the figure.

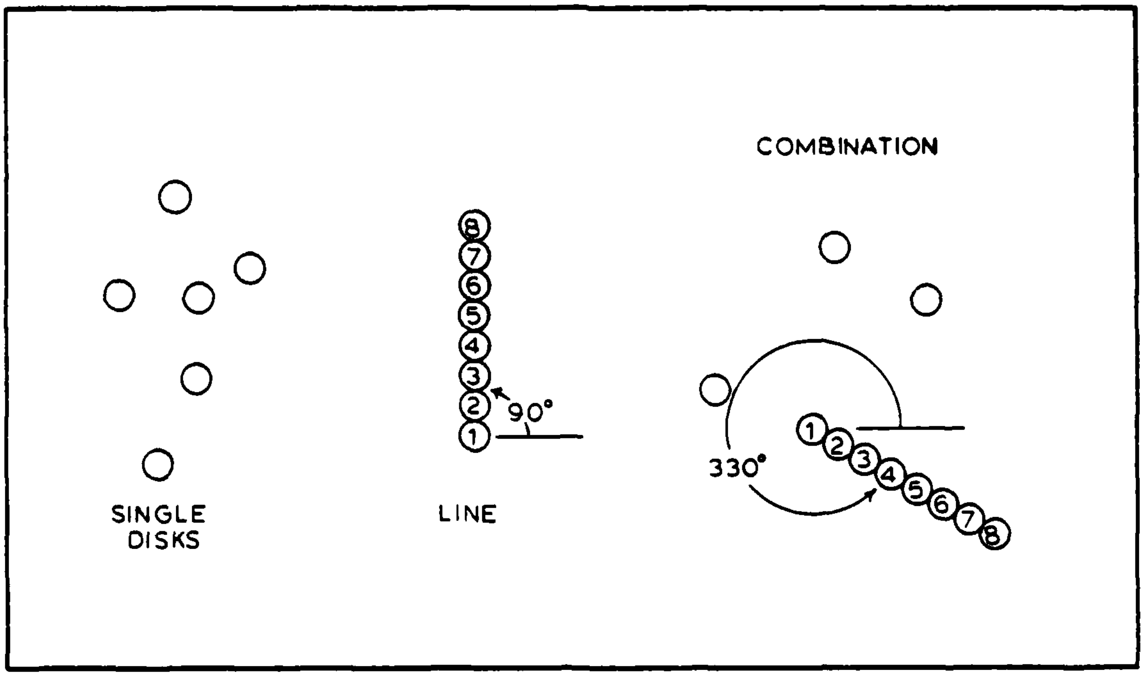
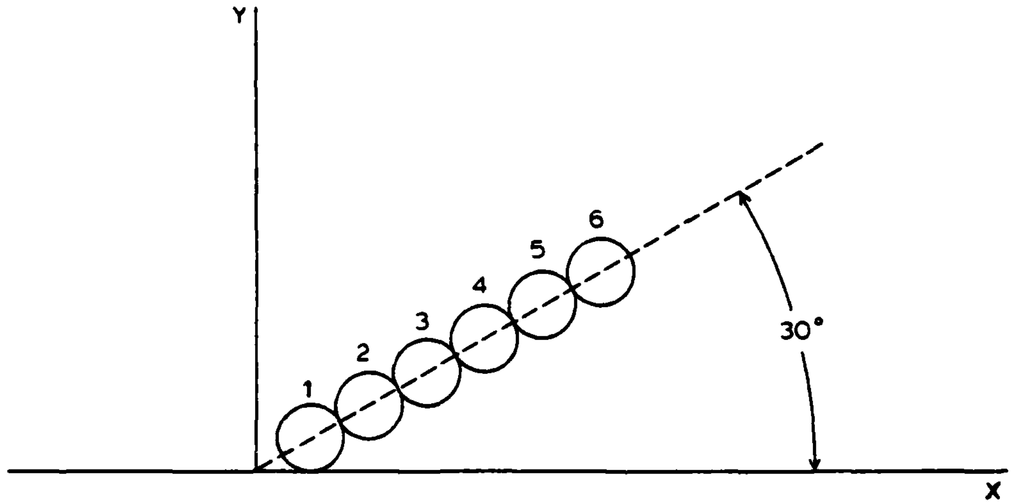


Fig. 3.2. Two Dimensional Substrate Examples



Prompt	Response
Buffer Editor Menu	A
Adding data on line 1	
Number of disks?	1
X coordinate	.5
Y coordinate	.5
Adding data on line 2	
Number of disks?	5
Angle	30
Adding data on line 3	
Number of disks?	0

Fig. 3.3. Line Substrate with Appropriate Buffer Commands

After the data is entered in the buffer, the matrix manager may be directed to calculate disk locations as specified in the buffer and then placed in the xy field matrix. Single disks require no calculations for matrix placement. Disks which are part of line segments composed of N disks require disk location calculations. This calculation consists of determining N equidistant points on a directed line segment as specified in the buffer. Since this requires

simple algebra and trigonometry, this procedure will not be derived here. The equations used to find the x and y coordinates of the N disks are:

$$X(n) = X(n-1) + \sqrt{2}n\cos(a)$$

$$Y(n) = Y(n-1) + \sqrt{2}n\sin(a)$$

where

X(n) = x coordinate
Y(n) = y coordinate
X(n-1) = x coordinate of disk to the left
Y(n-1) = y coordinate of disk to the left
n = number of disk counting from the left
a = angle of directed line segment measured from the positive x axis in the counterclockwise direction.

Understanding of this equation and the significance of the $\sqrt{2}$ factor will become clearer after the next section.

The TFG Model and Computer Implementation

The TFG model is a two dimensional mathematical representation of a film segment. To understand the model and how the computer implements it, three aspects of the model are discussed here. They are field and disk mechanics, disk dynamics, and field wrap around.

Field and Disk Mechanics. The TFG model assumes a two dimensional field with an x axis length of XAXIS and a y axis length of YAXIS. In the computer, this field is composed of three subfields; the x field, y field, and an occupancy field. These subfields are used by the computer to store x, y, and occupancy data for each point in the model xy plane. Each subfield is represented by a matrix made of (XAXIS) x (YAXIS) unit cells. This is shown in Figure 3.4.

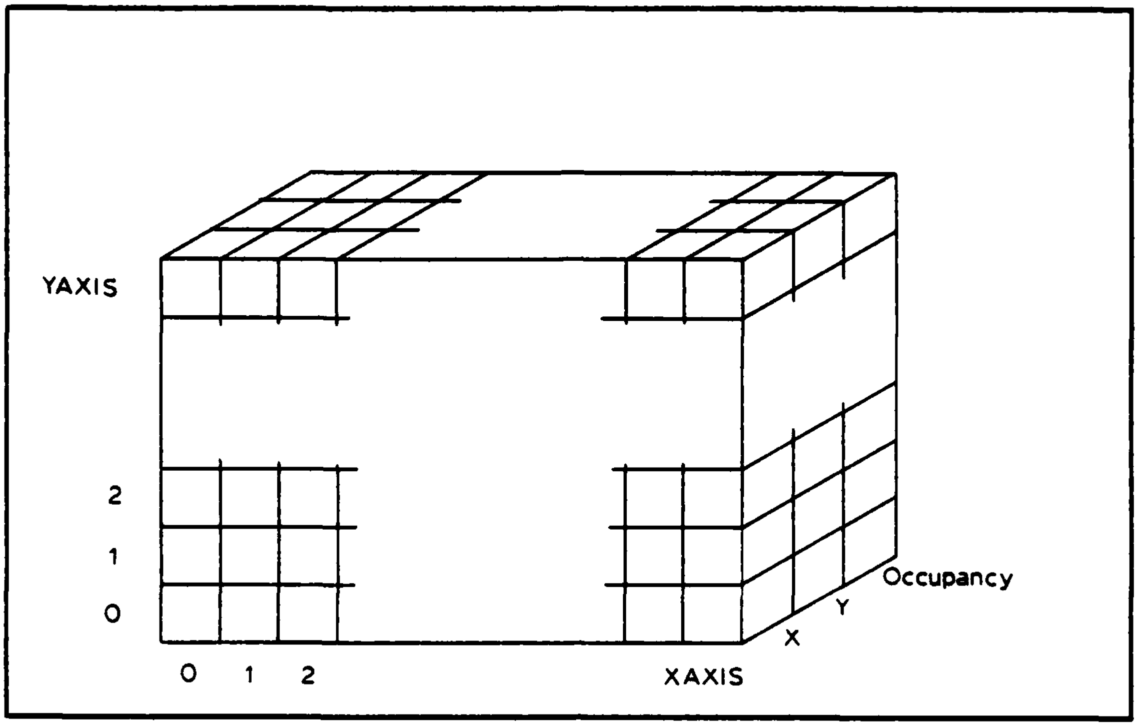
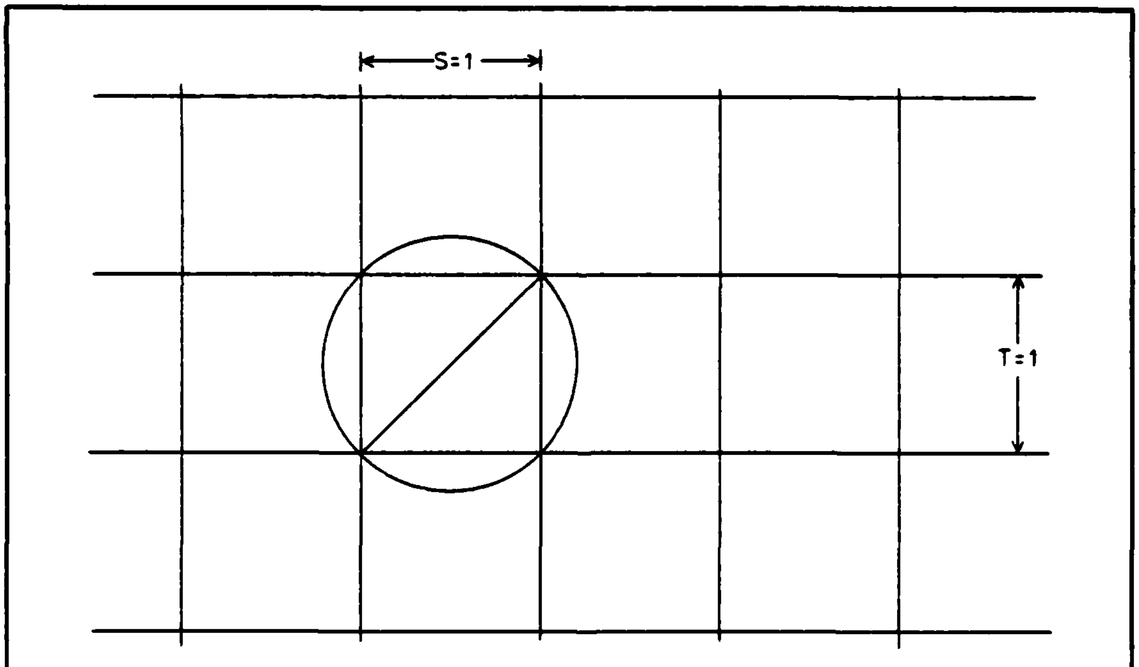


Fig. 3.4. Computer Subfields



$D^2 = S^2 + T^2 \qquad D = \text{diameter} = \sqrt{1 + 1} = \sqrt{2}$

Fig. 3.5. Disk Size

Particles within the xy plane are represented in the model by hard disks. All disks are of the same diameter and the diameter is such that only one disk can fit in a unit cell, i.e. the disk diameter is equal to the diagonal of a square unit cell. In other words, a disk diameter is equal to the $\sqrt{2}$ (See Figure 3.5). Disks are therefore represented in the computer by a disk radius and center coordinates. The computer stores disk coordinates in the matrix (See Figure 3.4) by storing each coordinate in the unit cell of the corresponding field. Unit cell addresses are made by truncating the coordinates. For example, consider a disk at location (4.3721, 25.9125). After truncation, this would become (4,25). The matrix cell assigned to the x and y coordinates respectively would be cell (4,25,1) and (4,25,2). Thus, 4.3721 would be stored in cell (4,25,1), 25.9125 would be stored in cell (4,25,2), and a 1 would be stored in cell (4,25,3) to indicate valid data in (4,25,1) and (4,25,2). The occupancy field can only contain a 0 or 1. A 0 indicates that the cell is unoccupied and any data contained by that cell in the x or y field is to be ignored. A 1, on the other hand, indicates that the cell is occupied and the corresponding x and y field cells contain good data.

Disk contact, as shown in Figure 3.6, is accomplished by placing the disk center one radius from a surface. It is not accomplished by checking for an intersection of the disk surface with some other surface. Likewise, two disks come into contact when their centers are the $\sqrt{2}$ or one diameter

apart. Finally, a third disk may not pass between two other disks unless the centers of the two other disks are separated by more than two disk diameters.

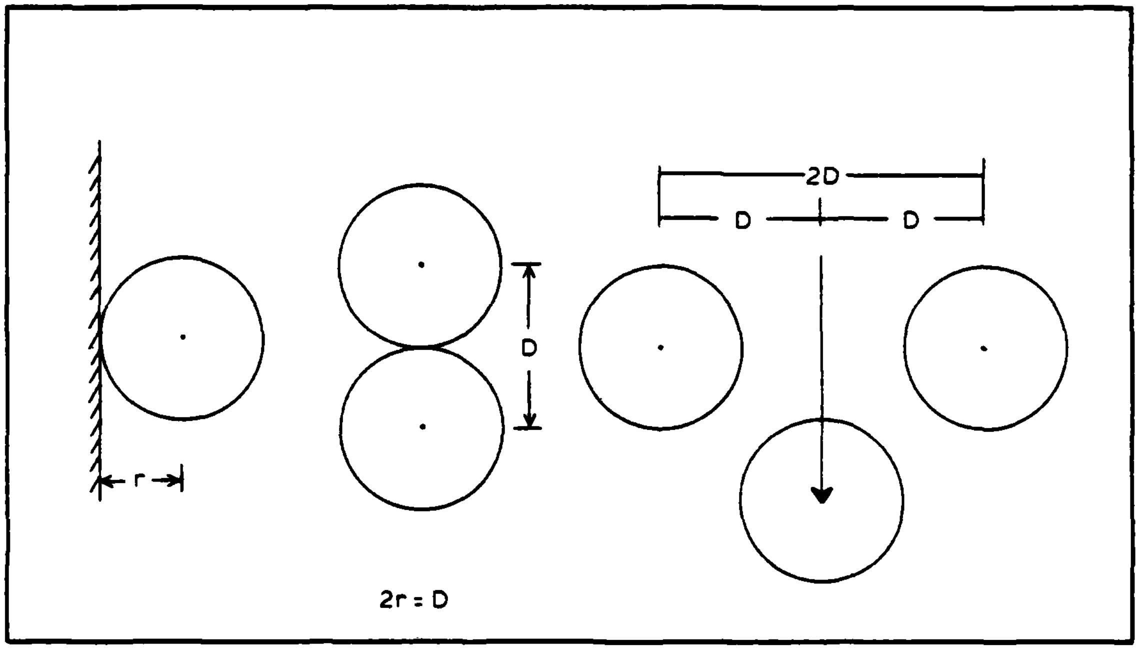


Fig.3.6. Disk Contact

Disk Dynamics. In the TFG model, incident hard disks travel within the field on a straight line which is inclined at an angle A from the y axis. The values of the x coordinate at which this line intersects the x axis are chosen randomly. Each hard disk travels on the straight line until it comes in contact with one of the already deposited hard disks. Furthermore, the incident disk always remains in contact with the disk in the film with which it first made contact (base disk). After collision the incident disk relaxes or moves along the base disk perimeter until it reaches the nearest "pocket" where it makes contact with a disk which has been previously deposited (See Figure 3.7).

The disk dynamics described here and pictured in Figure 3.7 are implemented in the computer through two algorithms. The first is the collision point determination algorithm. It tracks the particle in and calculates where the incident disk center is located upon collision. The second is the roll and rest point determination algorithm. Its function is to locate the nearest rest point and calculate where the incident disk center will be when it comes to rest.

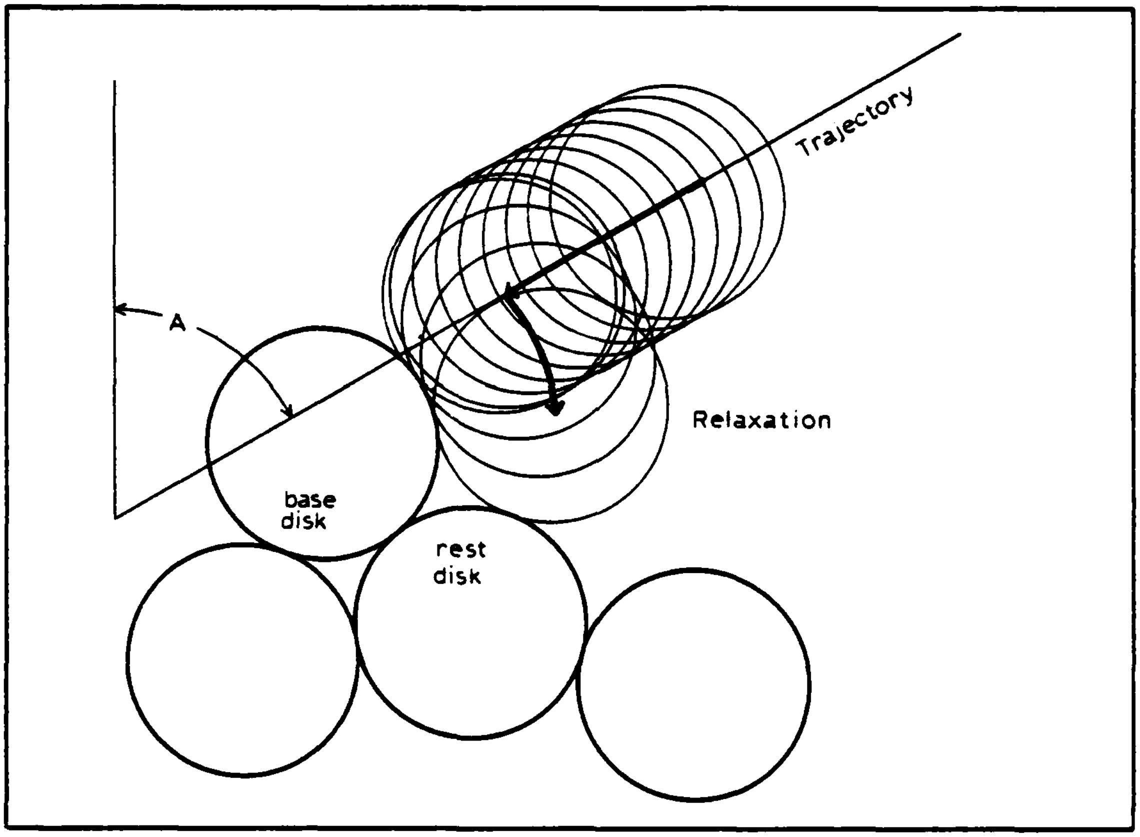


Fig. 3.7. Disk Contact and Relaxation

Collision Point Determination. There are many ways of implementing the tracking of a particle through its trajectory and simultaneously scanning the surrounding area for possible collision partners. The most obvious way would be

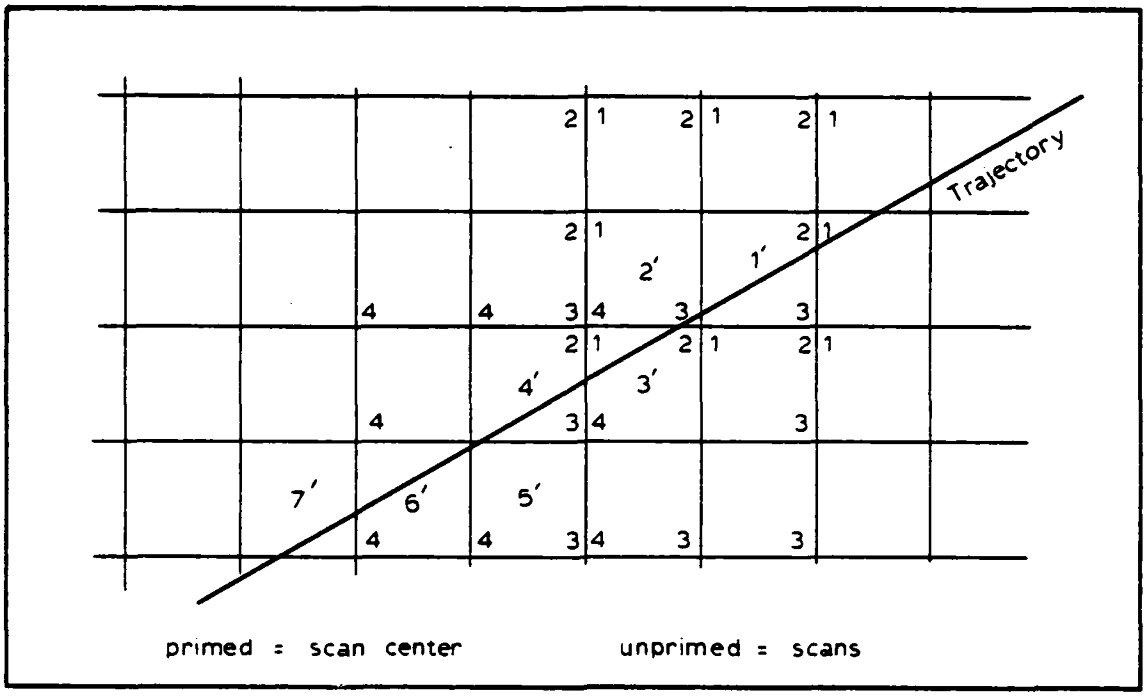


Fig. 3.8. Ordinary Area Scan

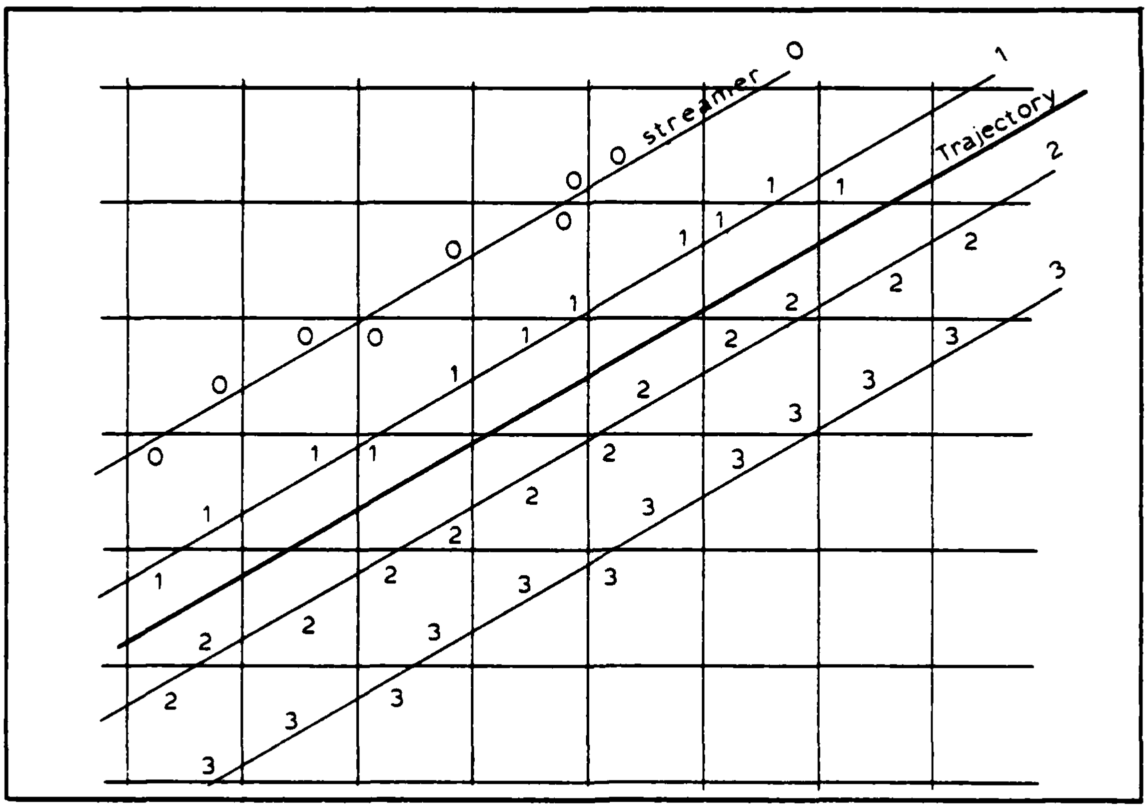


Fig. 3.9. TFG Model Scan

to start at the top of the field, move down the trajectory a step, scan for any particle within collision distance, move down another step, scan again and so-on (See Figure 3.8). This method, although easy to understand, is very inefficient. The program used and presented in Appendix A eliminates a lot of this methods inefficiencies.

The TFG program used reduces the inefficiency by two separate variations in the implementation of the model. First, the program stores and constantly maintains what the highest y cell value is. Trajectory tracking and collision scanning start one unit cell above this y value and therefore eliminate needless tracking and scanning for base cells. Second, the scanning technique used to find base cells is not an area search after each step as shown in Figure 3.8. This method would check the occupancy of a number of given cells many times. Instead, as shown in Figure 3.9, four streamers parallel to the trajectory search a "collision corridor" for occupancy in any cell that they contact. The streamers scan continues until encountering an occupied cell not previously encountered by another streamer. Concurrent with streamer scanning, occupied cells encountered are compared to see which would be the collision partner or base disk. Base disk determination is accomplished by trying all candidate base disks to see which yields the largest incident disk y coordinate.

Three items need to be expanded upon before moving on. The first is the term "collision corridor". From the

discussion on disk mechanics, it is known that a disk must be one disk diameter away from another disk for them to be in contact. Therefore, in order for a disk to be a collision partner with the incident disk, the disk must be within one diameter from the incident disk trajectory line. The area swept out by the perpendicular one diameter distance on both sides of the trajectory line is the collision corridor.

Next is the streamer spacing. Perpendicular spacing between streamers is always constant and equal. The spacing is chosen so that no matter what angle the streamers are at, no unit cell can be placed between them. At the same time the number of streamers should be as small as possible so that efficiency is kept high. Since the collision corridor is 2D thick or $2\sqrt{2}$ (see Figure 3.6), four streamers are required to ensure coverage. Their perpendicular spacing is $2\sqrt{2}/3$. The x axis intercepts of these streamers can be determined from,

$$S = a + \frac{2\sqrt{2}n}{3\cos(A)}$$

where

a = x coordinate of the left most streamer
A = deposition angle measured from substrate normal
n = streamer number starting with 0 and counting to the left (see Figure 3.9)

and

$\frac{1}{\cos(A)}$ = x axis correction for A other than zero.

Finally, the incident disk center coordinates are found by calculating the intersection of the trajectory line

equation and the one diameter contact circle. This is shown in Figure 3.10 and the equations for the coordinates are:

$$Y = \frac{-\left(\frac{(a-h)}{m} - k\right) + \sqrt{r^2\left(\frac{1}{m^2} + 1\right) - \left((a-h) + \frac{k}{m}\right)^2}}{\left(\frac{1}{m^2} + 1\right)}$$

$$X = \tan(A)\,Y + a$$

where

a = x intercept of trajectory line
h = x coordinate of the circle center
k = y coordinate of the circle center
A = deposition angle from substrate normal
m = tangent of A
r = disk radius

The derivation (See Appendix B) of these equations consists of simple algebra and trigonometry.

<u>Roll and Rest Point Determination</u>. Once the base disk is determined, the incident disk will roll around the base disk perimeter until it makes contact with a third disk. This third disk or rest disk is determined by finding that disk which allows the incident disk to move the shortest distance (See Figure 3.7). Then the final location of the incident disk is recorded in the field.

The program implements this by searching a seven by seven unit cell area around the base disk unit cell. All occupied cells which contain disk center coordinates within a two disk diameter radius of the base disk center are considered as possible rest disks. Then all of the rest position coordinates given the possible rest disks are found. The distances from the collision point to all possible rest

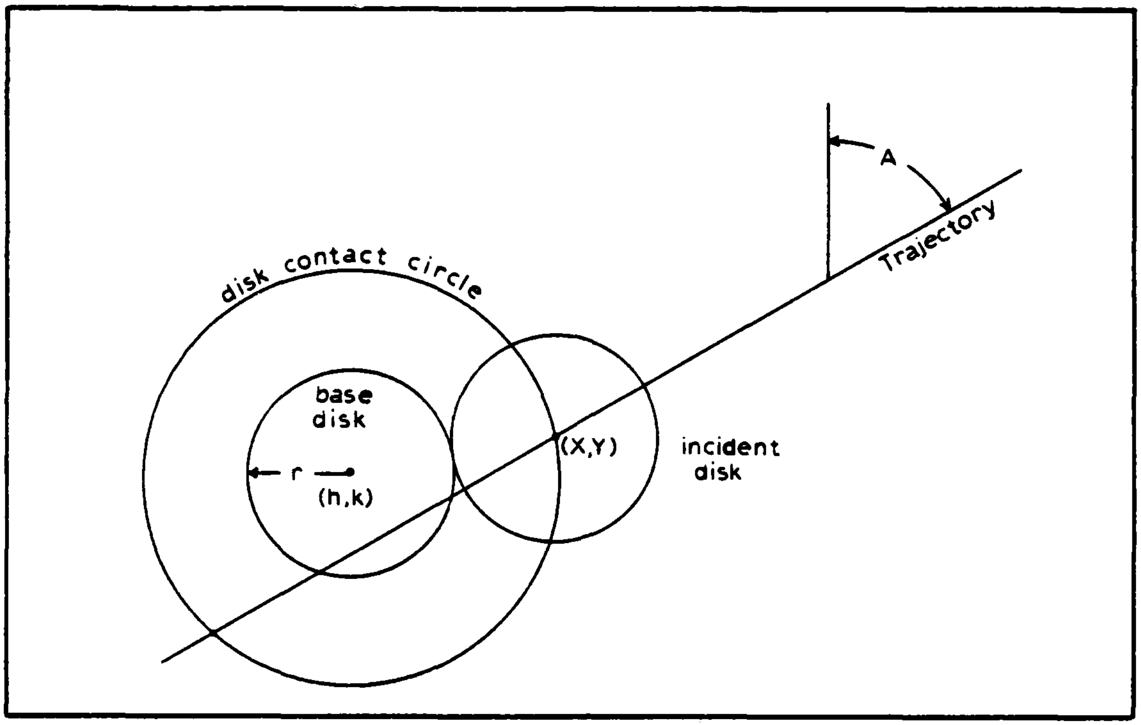


Fig. 3.10. Collision Point Determination

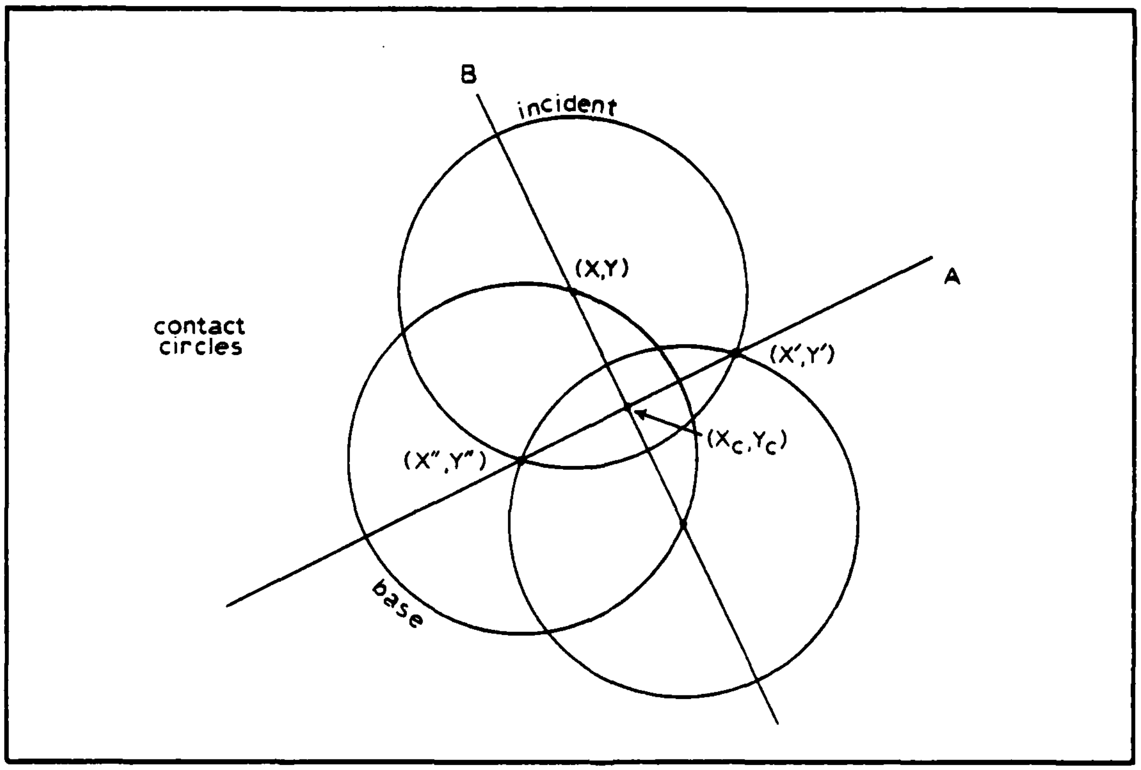


Fig. 3.11. Roll and Rest Point Determination

points are calculated and the shortest selected. Those rest point coordinates are then stored in the field matrix.

The formulas for finding the rest points are actually quite simple. They are:

$$X = \frac{X'R \pm \sqrt{X'^2R^2 - R(R^2 - 8Y'^2)}}{2R}$$

$$Y = \frac{Y'R \pm \sqrt{Y'^2R^2 - R(R^2 - 8X'^2)}}{2R}$$

where

$$R = (X'^2 + Y'^2)$$

and

X' = x separation of base and rest disk centers
Y' = y separation of base and rest disk centers.

The derivation of these formulas (See Appendix B) is based on the fact that the coordinates of the final resting position must be a distance of one disk diameter from the base and rest disk centers. This means that the coordinates of final resting position must be the center of a circle which passes through the base and rest disk centers. This is shown in Figure 3.11.

<u>Field Wrap Around</u>. In the TFG model, a field wrap around is provided. This makes the field periodic. So when film growth extends beyond the right boundary it is shifted back into the field on the left side. This makes more efficient use of the computer memory available and provides a means of recapturing data lost when it extends beyond the right boundary.

All data written in the buffer is also written in the first 10 locations and vice versa.

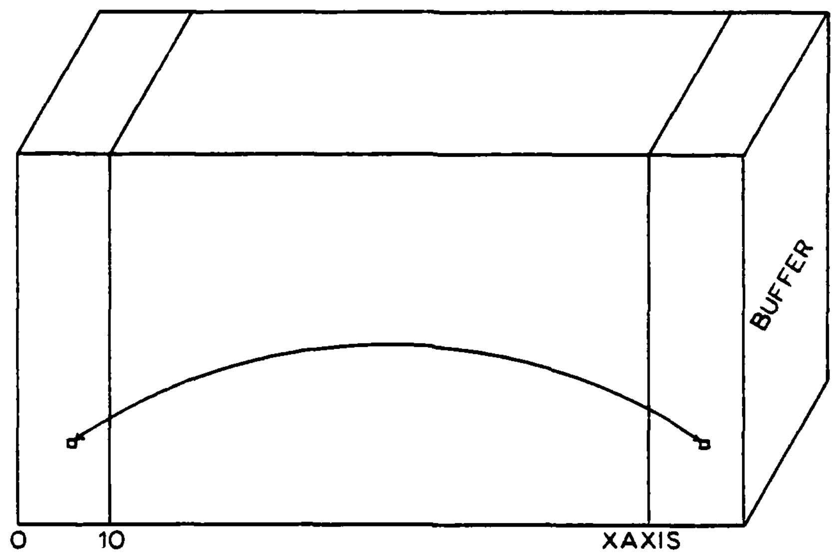


Fig. 3.12. Field Wrap Around Buffer

The implementation of the wrap around feature was accomplished by providing a ten unit buffer extended off the right side of the field (See Figure 3.12). This buffer is used simply to keep the collision point determination and roll and rest point determination algorithms simple. The buffer serves no other functional purpose. If the buffer were not provided, additional algorithms would have to be added to maintain synchronization between cells on the right side of the field and cells on the left. For example, in the roll and rest point determination algorithm, a seven by seven unit cell area around the base disk is searched for possible rest disks. If the base disk is centered on a y axis unit cell,

and no buffer is provided, part of the search pattern would be shifted to the left side of the field. However those cells on the left side of the field could not produce any possible rest disks, since they would be outside the two disk diameter radius. Consequently, film growth would be stopped at the right boundary. The only correction to this can be provided by a shifting correction algorithm within the rest point determination algorithm or a simple buffer.

TFG Analyzer

The TFG Analyzer provides the TFG simulator a means of determining the density, relative density, and the angle of columnar growth. It consists of two subprograms. One of the subprograms calculates field densities where the other calculates angle correlation numbers to be used in angle analysis.

Density. The theory of operation for density analysis is straight forward. Analysis of the density is not performed on the exact x and y coordinates but on the occupancy held matrix data. The density is found by

$$\mathrm{DEN} = \mathrm{DSKCNT}/\mathrm{CELCNT}$$

where

DSKCNT= the number of occupied cells
CELCNT = the total number of cells in the area being analyzed.

In addition to standard density, a relative density is also determined (see Appendix B). This is a density relative to hexagonal packing in two dimensions and is given by

$$\mathrm{RDEN} = \sqrt{3}\,\mathrm{DEN}.$$

<u>Angle</u>. The theory of operation for angle correlation is a little more involved and builds off the density theories. In this subprogram, a correlation number is determined for all whole degree angles from 0 to 89. The angle with the largest correlation number is the angle of columnar growth.

The correlation number for a given angle is found in several steps. First, the relative density of a two column trajectory is calculated. Then a modified average deviation from the relative film density is found. If the line density is less than or equal to the relative density, the average deviation is calculated by:

$$S = [(RDEN - DENLIN)/(RDEN)]^2$$

otherwise

$$S = [(DENLIN - RDEN)/(1 - RDEN)]^2$$

where

RDEN = the relative density of the film
DENLIN = the relative density of the trajectory.

The trajectory is then slid over one unit cell and the average deviation is found again. This process is repeated until the whole x axis has been traversed. All of the average deviations are then averaged together and this becomes the correlation number.

IV Validation of the Computer Program

Validation of computer programming is essential if any amount of confidence in a program are to be gained. It is for this reason that the program was checked with great detail to ensure that the programming was doing what was intended. Validation of the computer program/model consisted of four separate tests. These are explained below and are presented in the order in which they were accomplished.

Test one was a simple verification that the interactive logic and input was functioning properly. It consisted of trying all possible commands listed in the programs various menus and entering selected data where required. For example, one sequence tried was entering data in the substrate data buffer by entering A, 1, .2, 1.5, 225, 0. This sequence should create two lines in the data buffer as follows:

NDSK	X Coord	Y Coord	Angle
1	.2000	1.5000	.0000
225	.0000	.0000	.0000

Test two was designed to verify the substrate data entry and substrate construction. It consisted of three separate scenarios which generated flat, triangular well, and square well shaped substrates. Each scenario produced a printout giving cell content after substrate construction. The cell content was then compared to hand calculated results. Printout coding used in this test are indicated in the Fortran listing in Appendix A by a CT2 in the left most columns.

Test three was designed to verify the particle trajectory, collision point determination, roll and rest point determination, wrap around, and matrix full routines. The test consisted of two phases. The first, used a special version of the TFG simulator which was modified to print out all its calculations in the above routines and its random number generator was primed to give 10 known numbers. These numbers produced incoming particles which tested the limits of many critical routines. The printouts of these 10 situations were then compared to calculated particle movement and rest points. The second phase also used a special version of the TFG simulator. This version was modified to check for cell overwriting and spacing between the incoming particle, the collision particle, and the rest particle. To ensure that even the most minute error in the model would be caught, 340,000 particles were deposited (20,000 disks at 17 random deposition angles every 5 degrees from 0 to 80 degrees). Again, applicable printout coding is indicated in Appendix A.

Test four was designed to verify proper operation of the analysis routines. Again, a special version of the TFG simulator was used. Here the simulator was modified to print out various steps in the analysis. These are indicated in Appendix A by CT4. In addition, a test case was then run and compared with expected results.

After some programming error correction, all items tested passed with no errors noted. It should also be noted that with testing and normal program operation no errors to

date have been found. This adds up to over .5 million disks deposited.

V TFG Simulator Results

Once a TFG simulator was made and tested, the next important step was to run various depositions and check the results with those of known experimental works. Therefore, analysis of nine depositions were made. They were at A equals 0, 10, 20, 30, 40, 50, 60, 70, and 80 degrees. With this selection of deposition angles; density dependence on A, microstructure, and angle of columnar growth can be compared with experimental results and observations presented in Chapter II.

It should also be noted that the analysis was made of data in a 320x240 matrix on rows between 20 and 170. The 150 unit cell analysis was done to eliminate density variations which have been observed at the bottom and top of the film.

Density. The relative density of the previously stated depositions are shown in Figure 5.1. Except for the 0 and 10 degree depositions, larger deposition angles produce lower density material. This small disagreement could easily be caused by density variations in rows near the upper and lower analysis bounds. Density has been observed in other deposition to vary slightly.

Microstructure. The microstructure plots in Figures 5.2-5.5 are high density plots with 15 to 25 thousand disks. Only four of the nine depositions are shown here, since the structures do not change dramatically over 10 degree

intervals. The plots shown are of 0, 30, 60, and 80 degree depositions.

Column Angle. Angle analysis plots of all depositions are presented in Figures 5.6-5.14. These are presented here in their entirety because the angles, as determined by the maximum correlation points, are not always conclusive. Some have secondary maximum points which could be the angle of columnar growth. In Figure 5.15, all the maximum and secondary maximum points are plotted on a graph with a plot of the tangent rule. The results are relatively close and again the variation could be caused by density variations. Only this time the density variations would have to be in the local area of the trajected line.

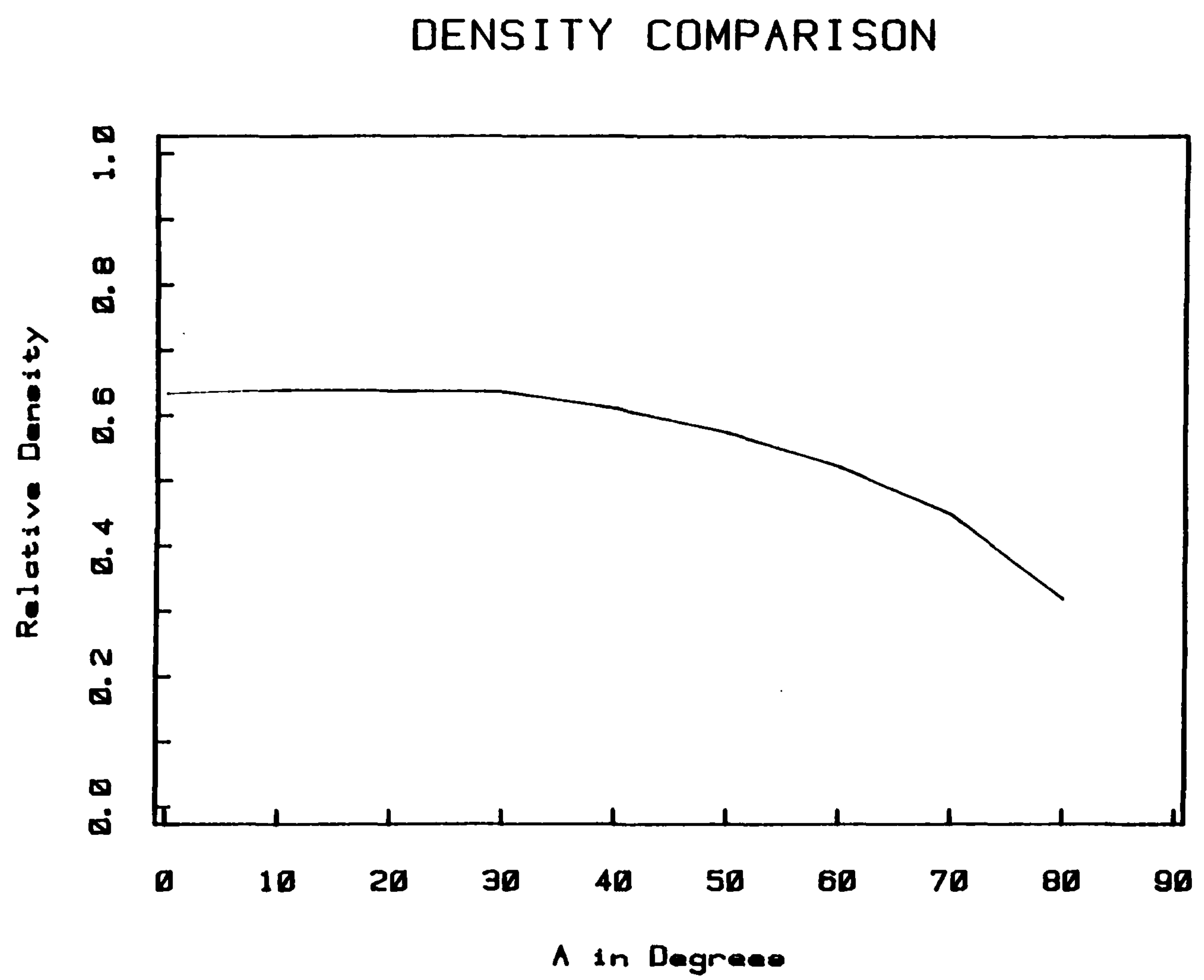


Fig. 5.1. Relative Density at Deposition Angles Less Than 80 Degrees

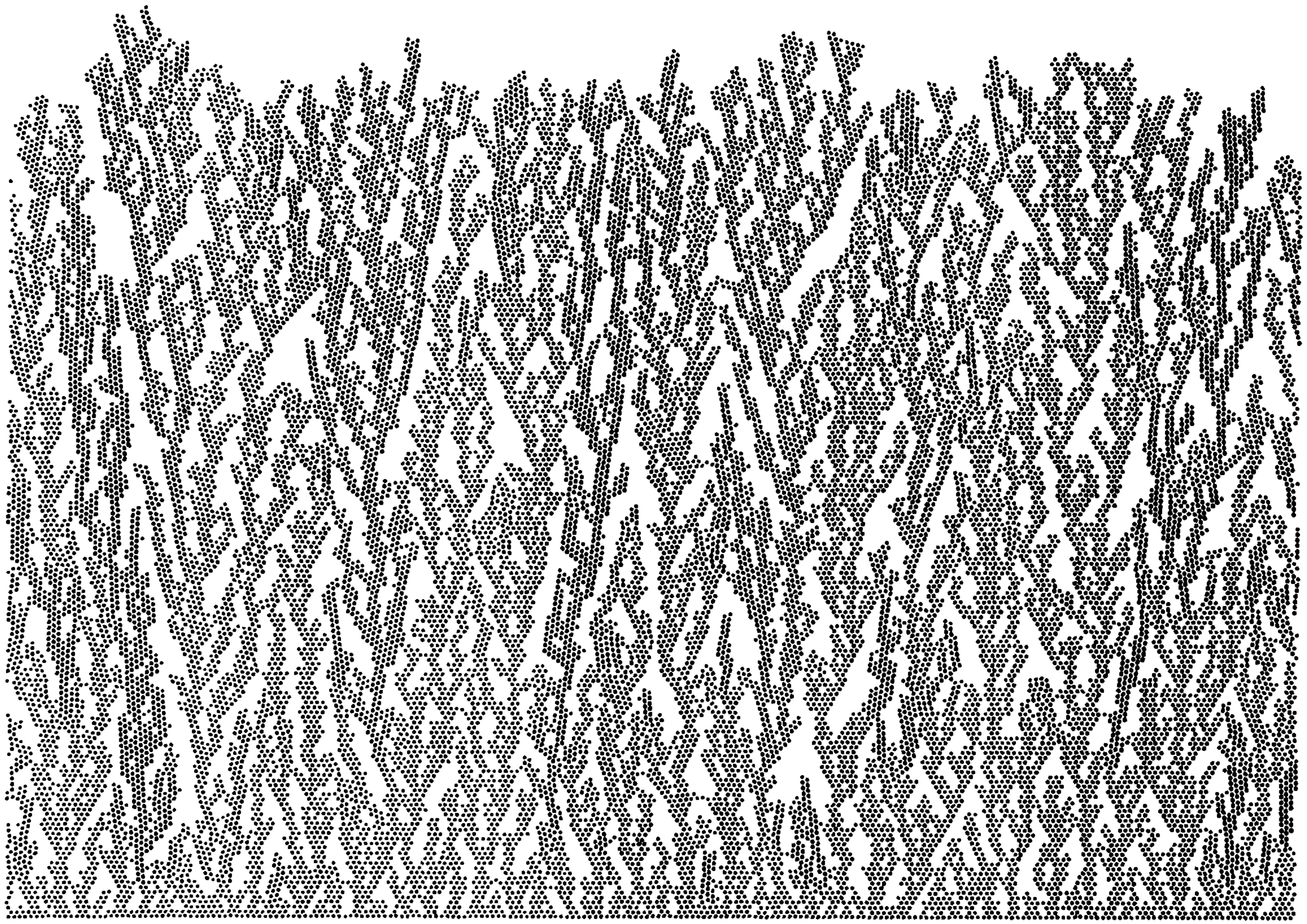


Fig. 5.2. Microstructure Form by Normal Incidence Deposition

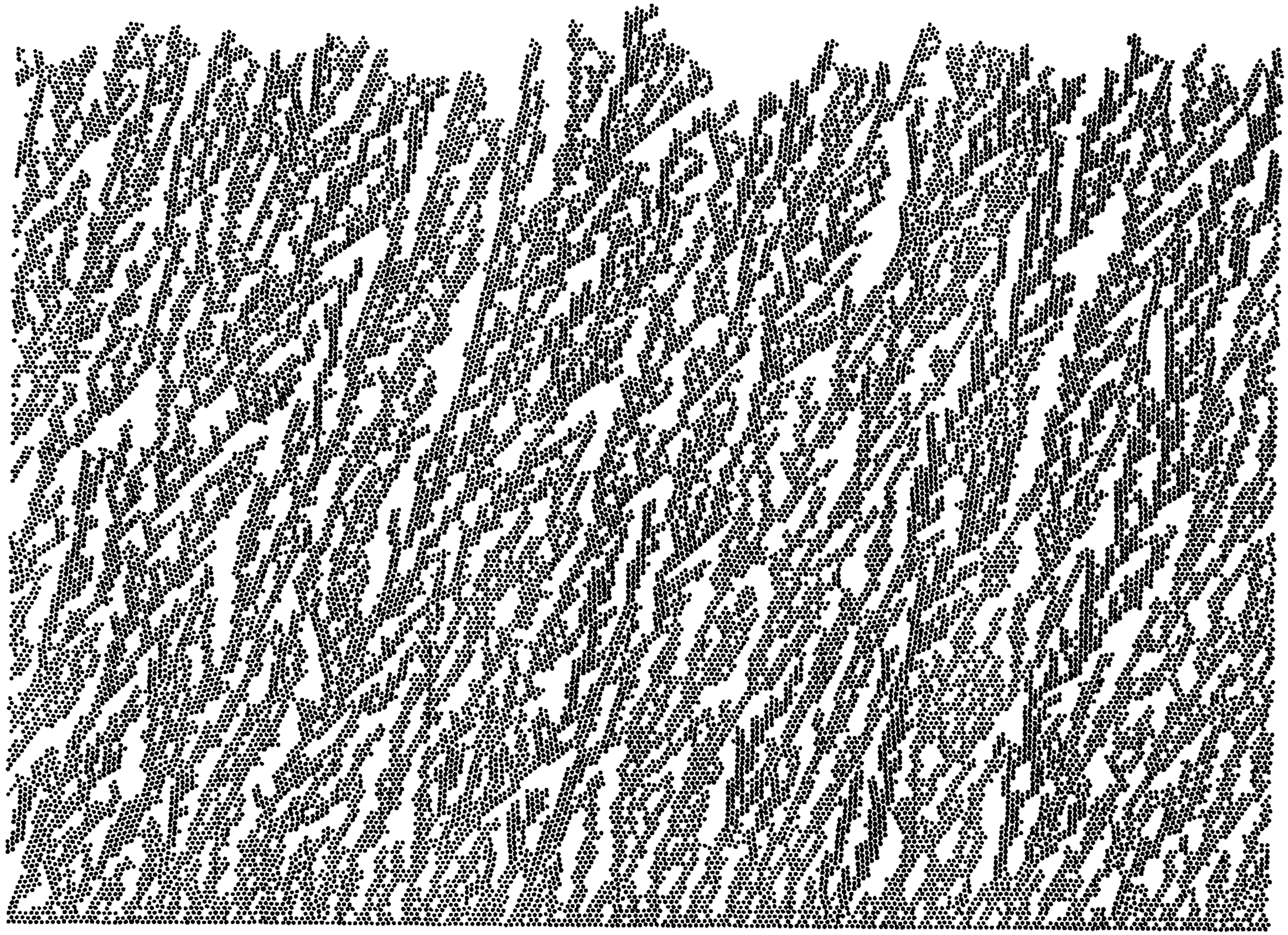


Fig. 5.3. Microstructure Form by 30 Degree Deposition

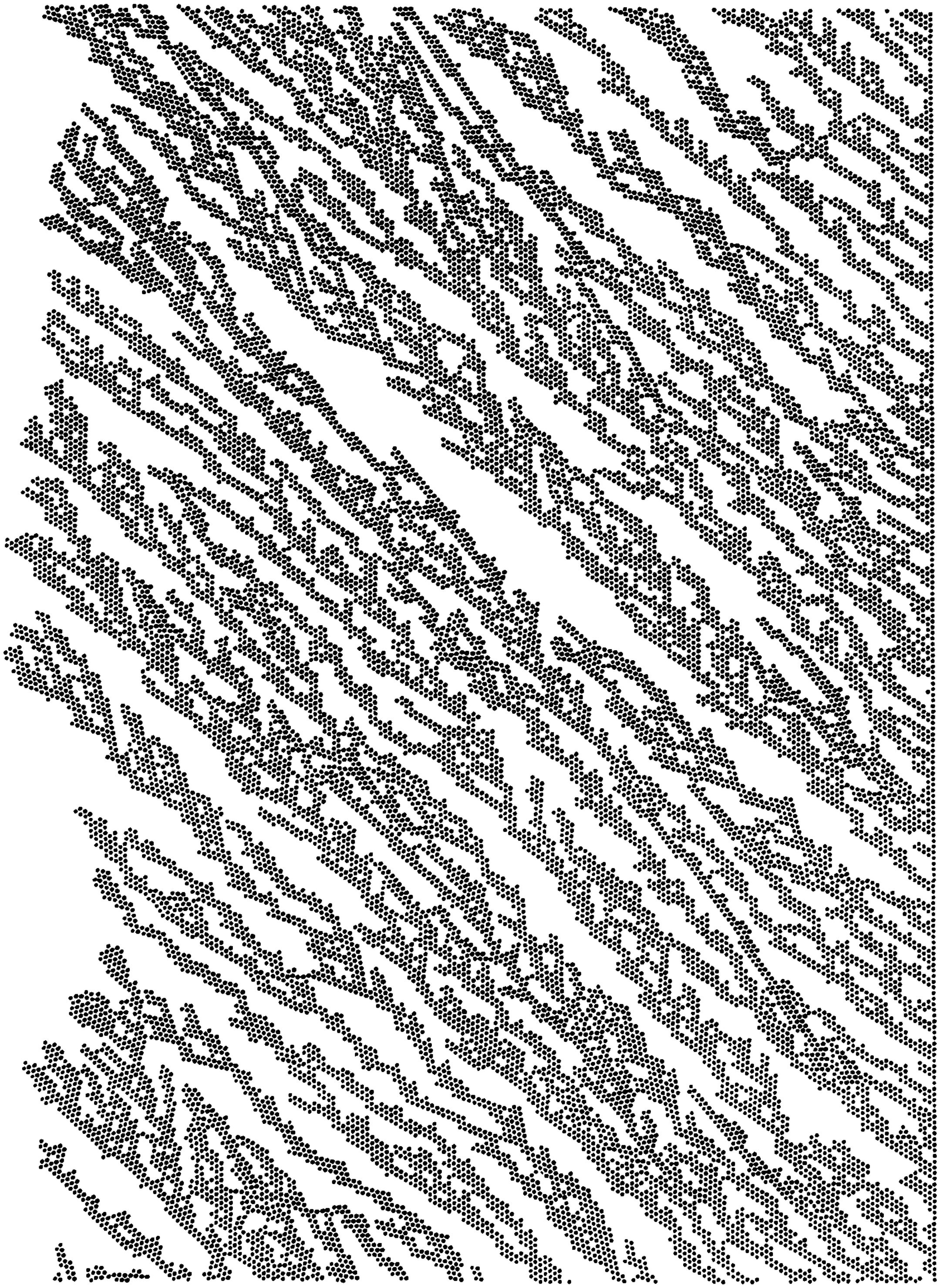


Fig. 5.4. Microstructure Form by 60 Degree Deposition

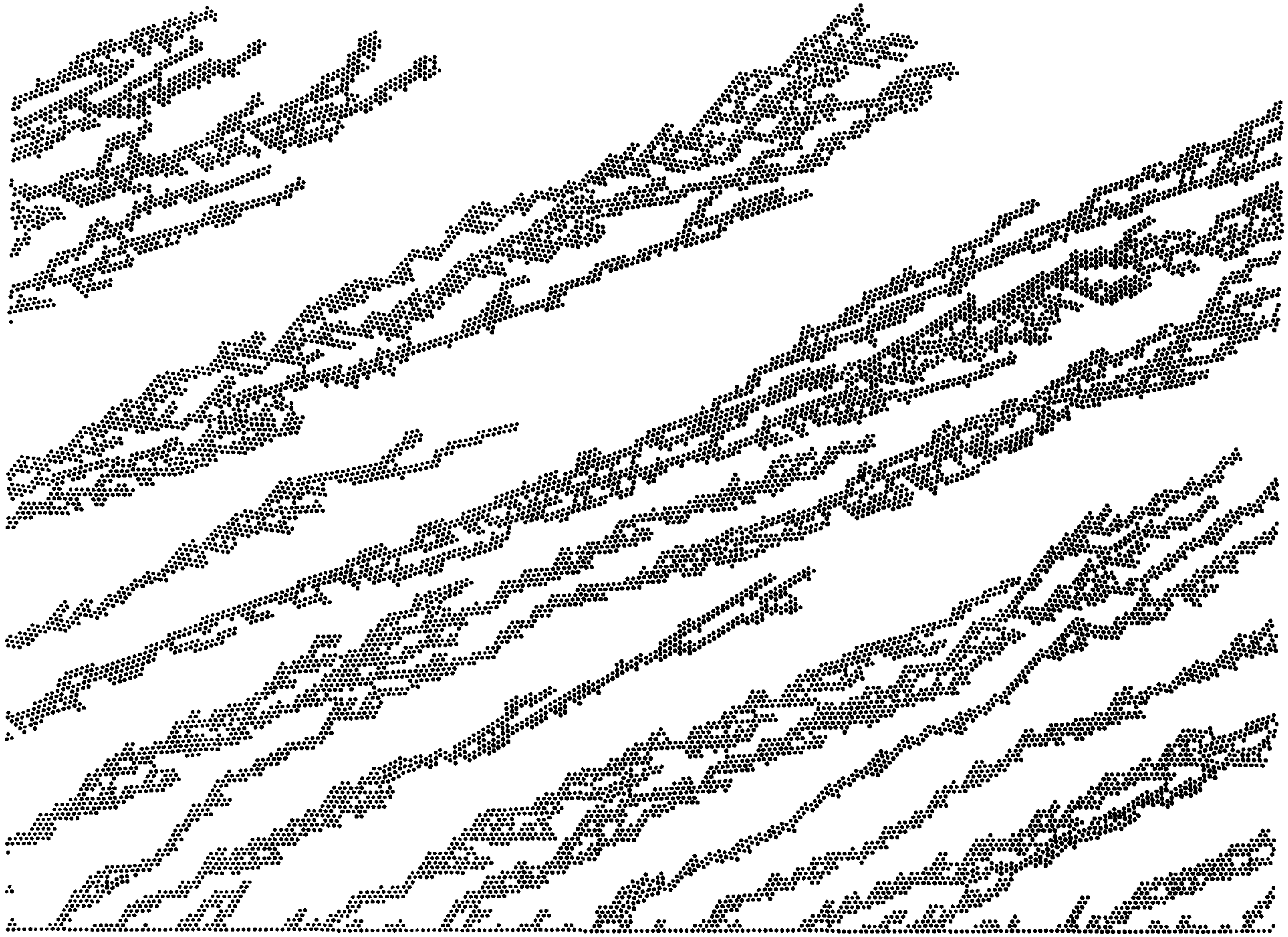


Fig. 5.5. Microstructure Form by 80 Degree Deposition

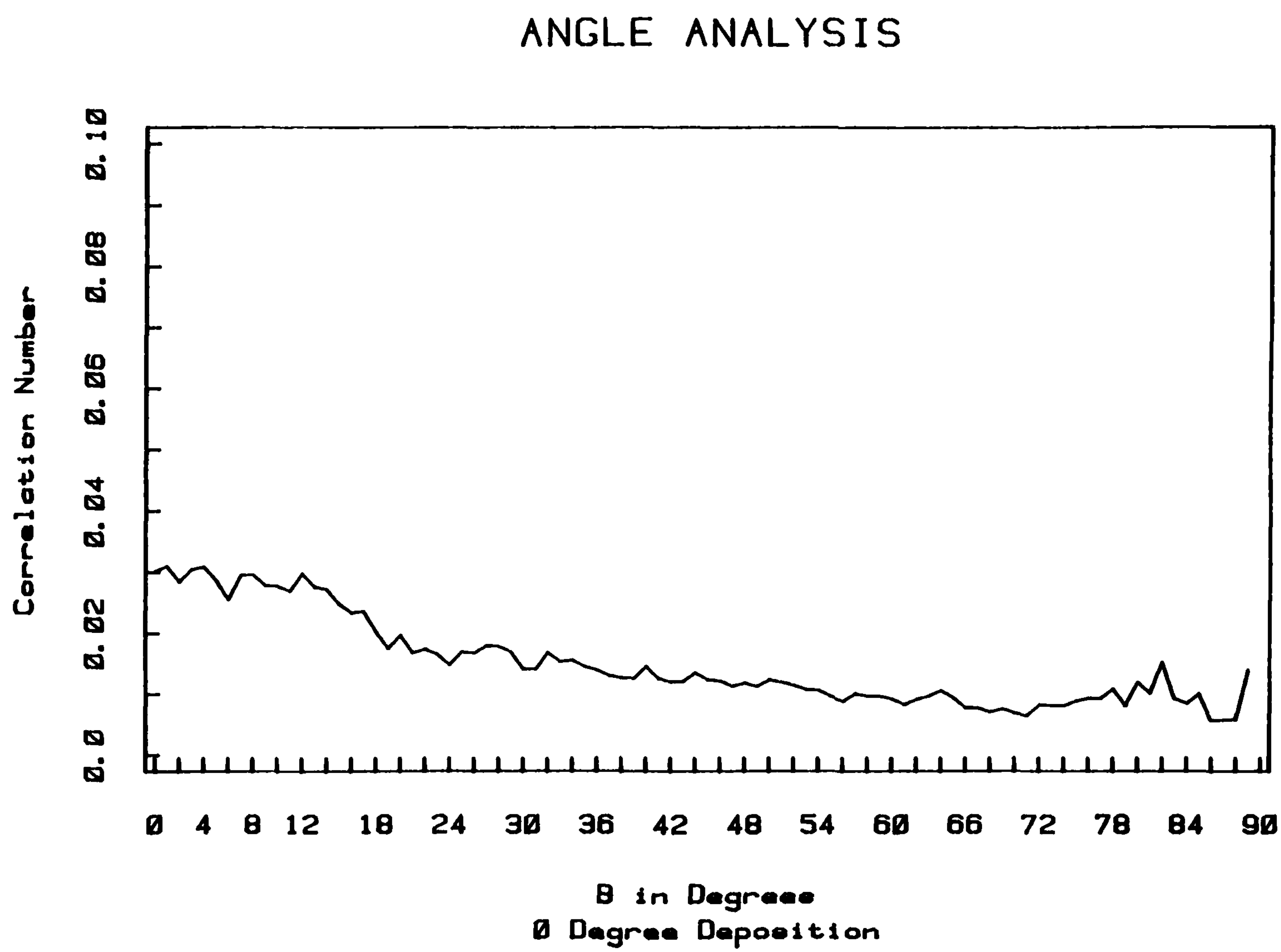


Fig. 5.6. Columnar Angle Analysis of Film Formed by Normal Incidence Deposition

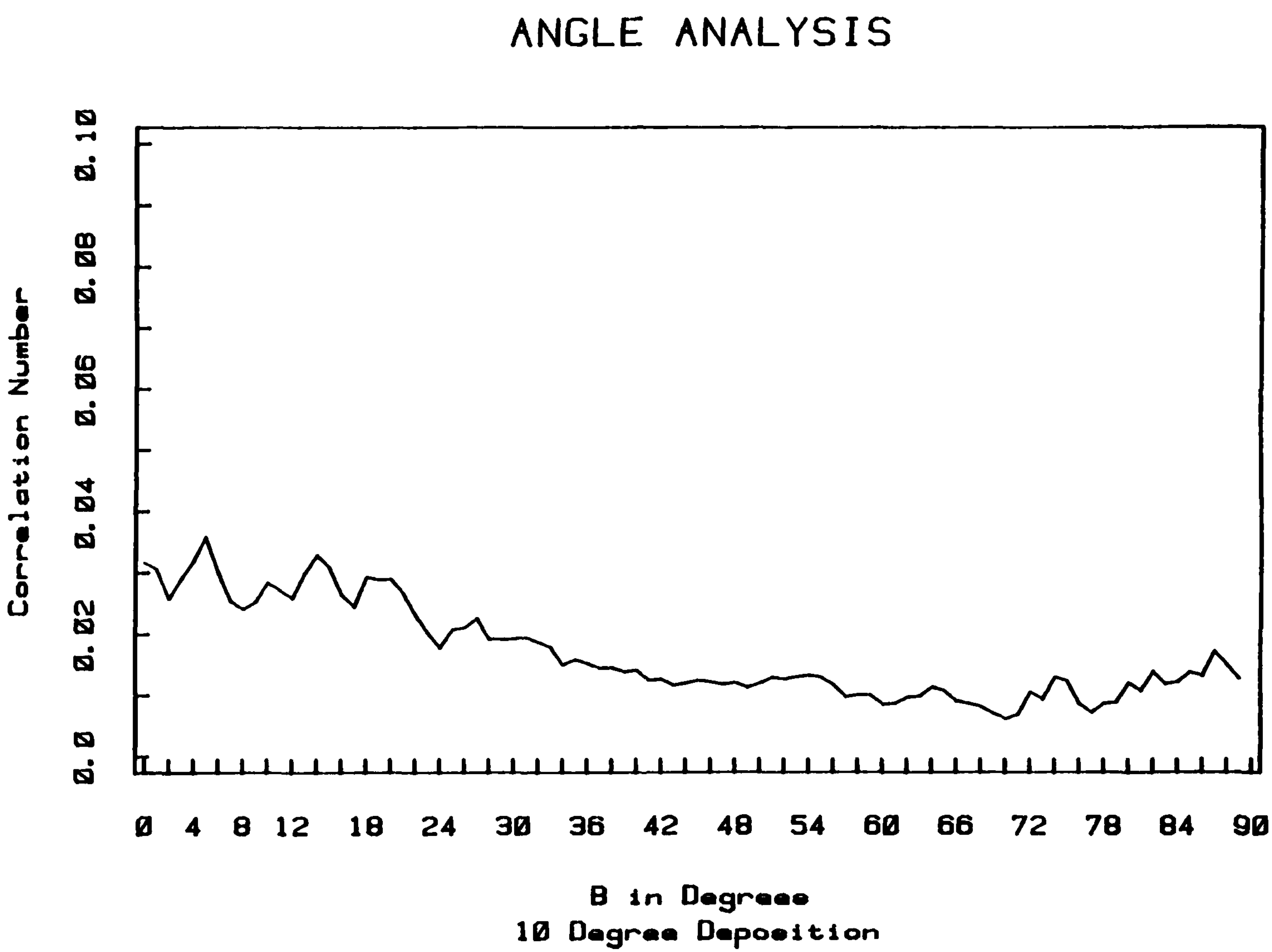


Fig. 5.7. Columnar Angle Analysis of Film Formed by 10 Degree Deposition

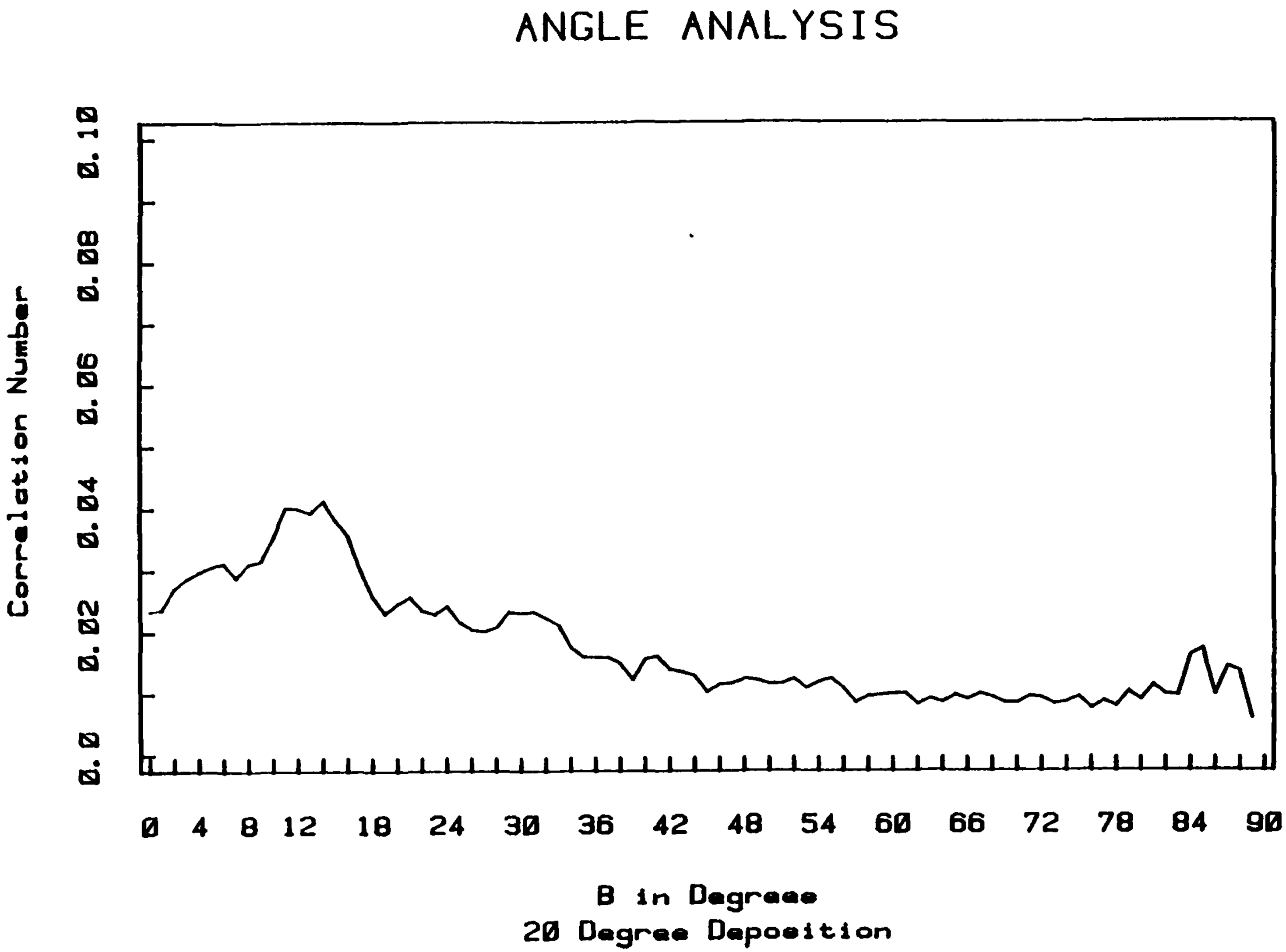


Fig. 5.8. Columnar Angle Analysis of Film Formed by 20 Degree Deposition

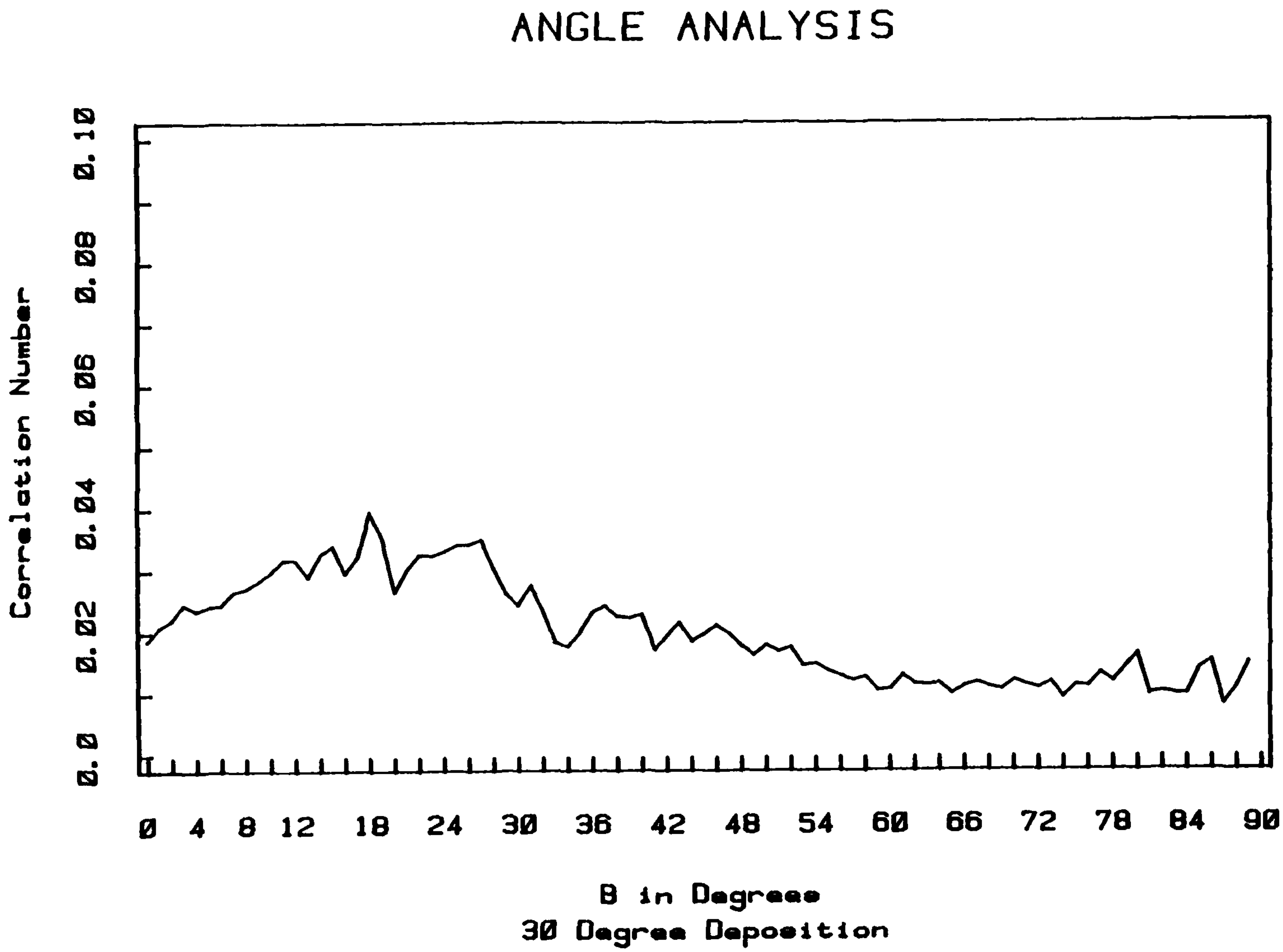


Fig. 5.9. Columnar Angle Analysis of Film Formed by 30 Degree Deposition

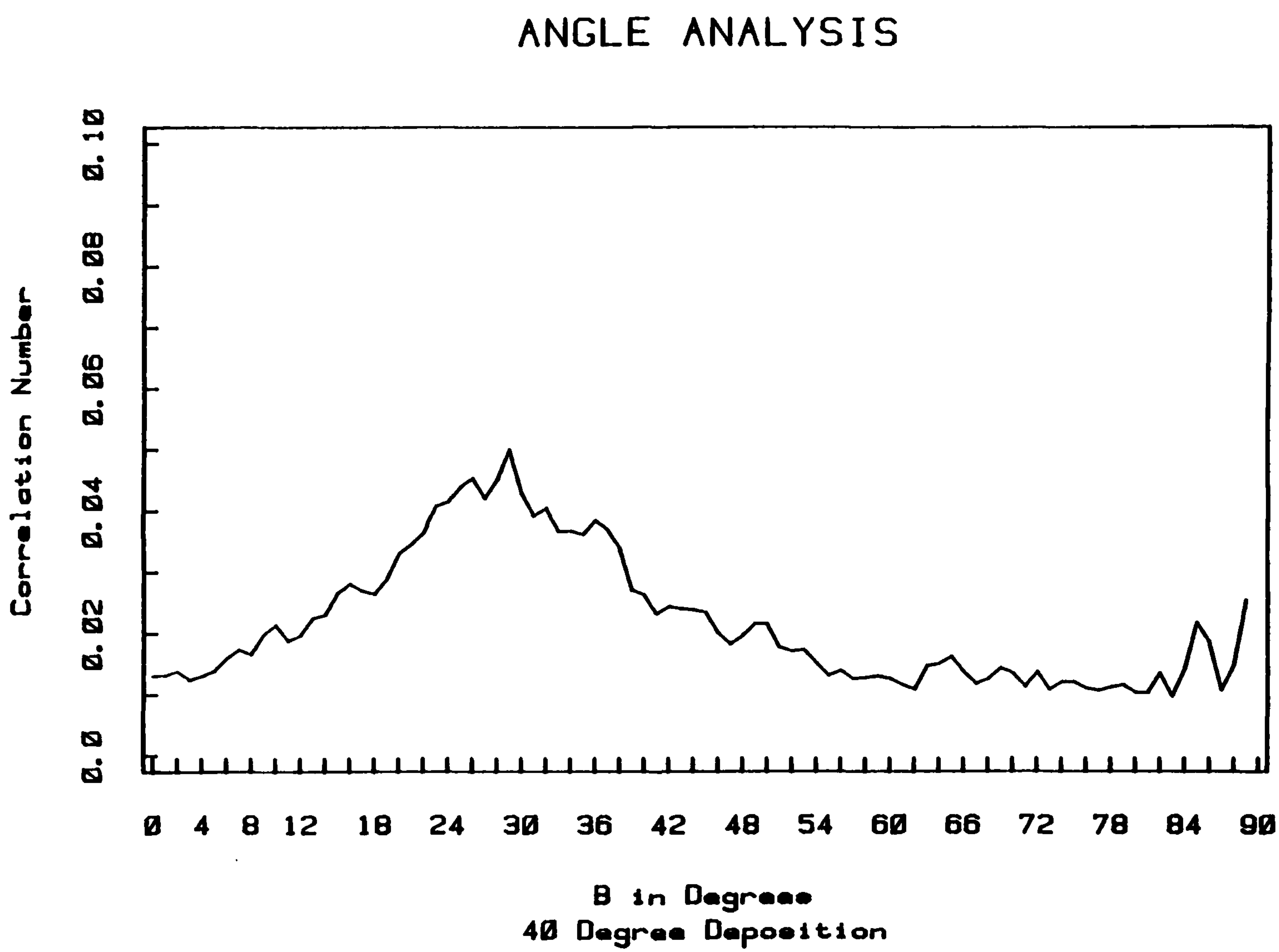


Fig. 5.10. Columnar Angle Analysis of Film Formed by 40 Degree Deposition

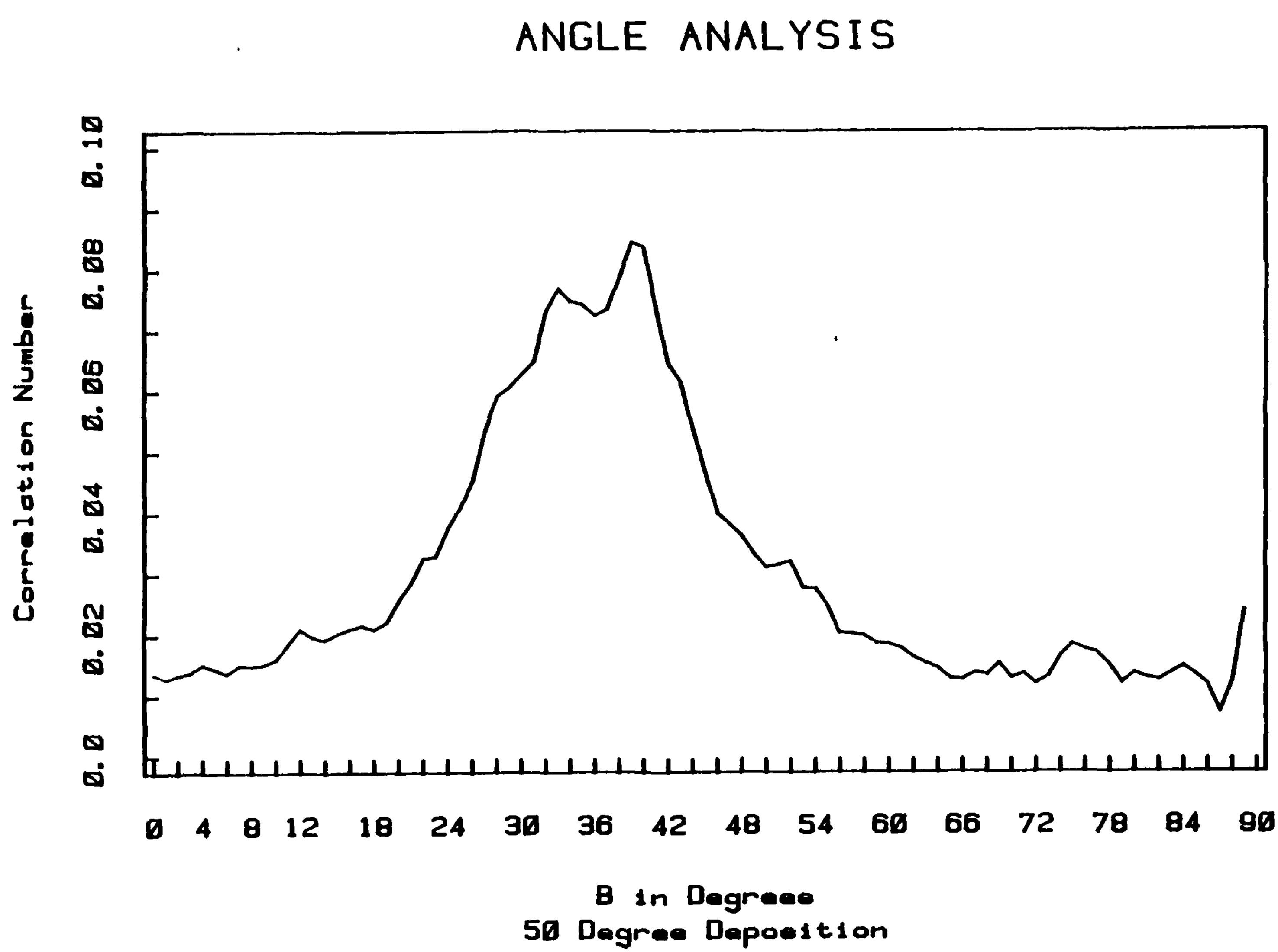


Fig. 5.11. Columnar Angle Analysis of Film Formed by 50 Degree Deposition

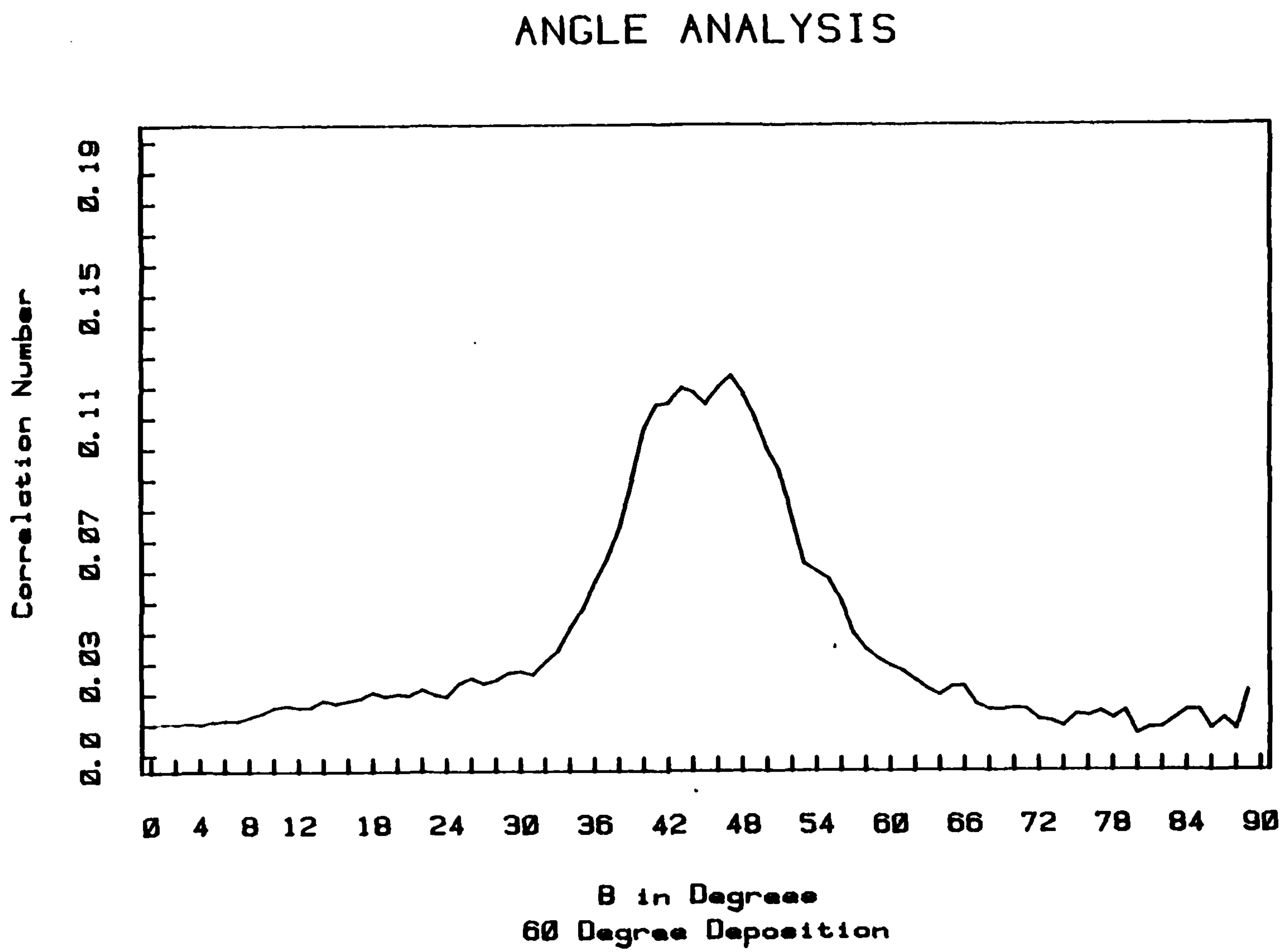


Fig. 5.12. Columnar Angle Analysis of Film Formed by 60 Degree Deposition

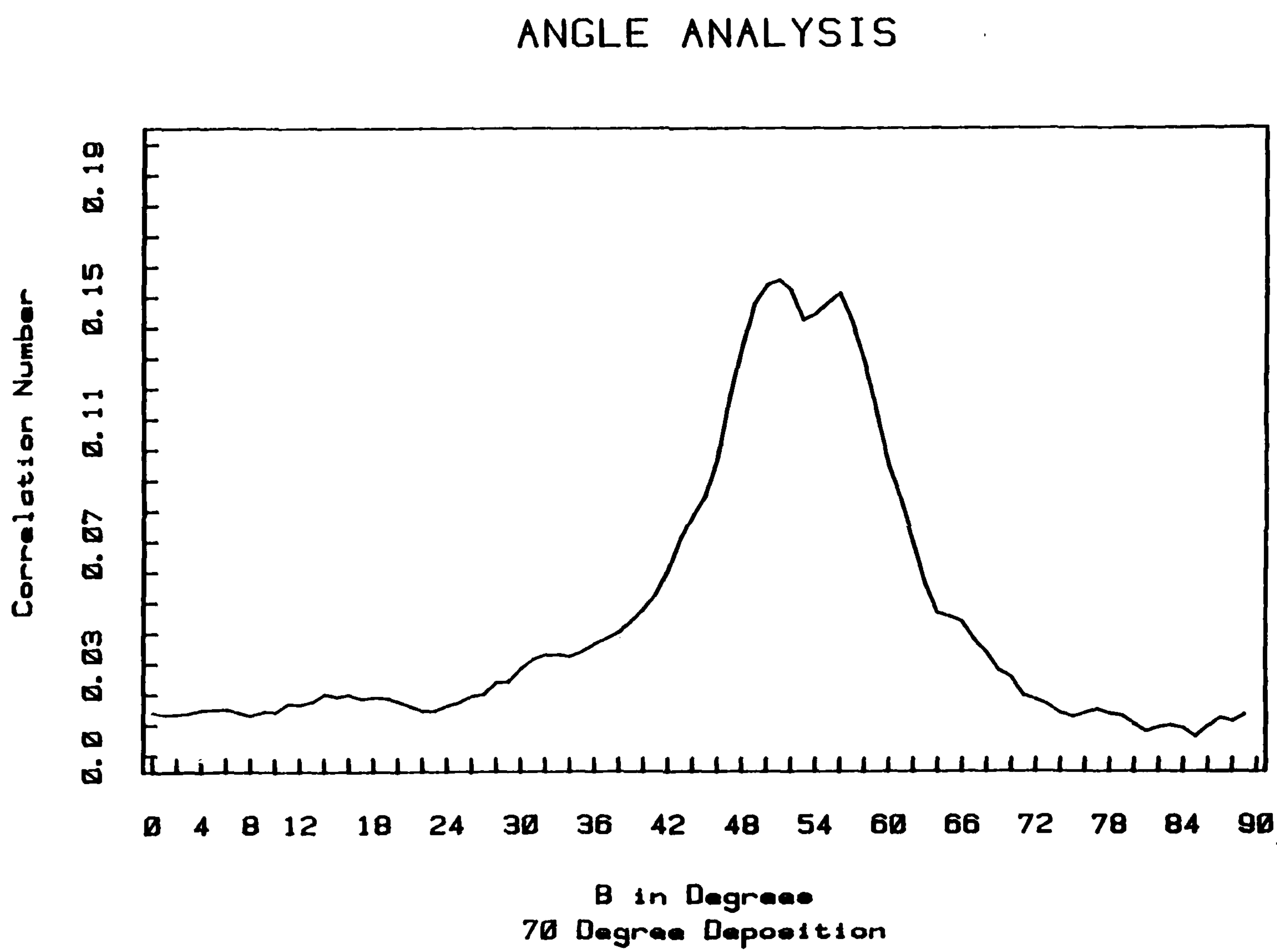


Fig. 5.13. Columnar Angle Analysis of Film Formed by 70 Degree Deposition

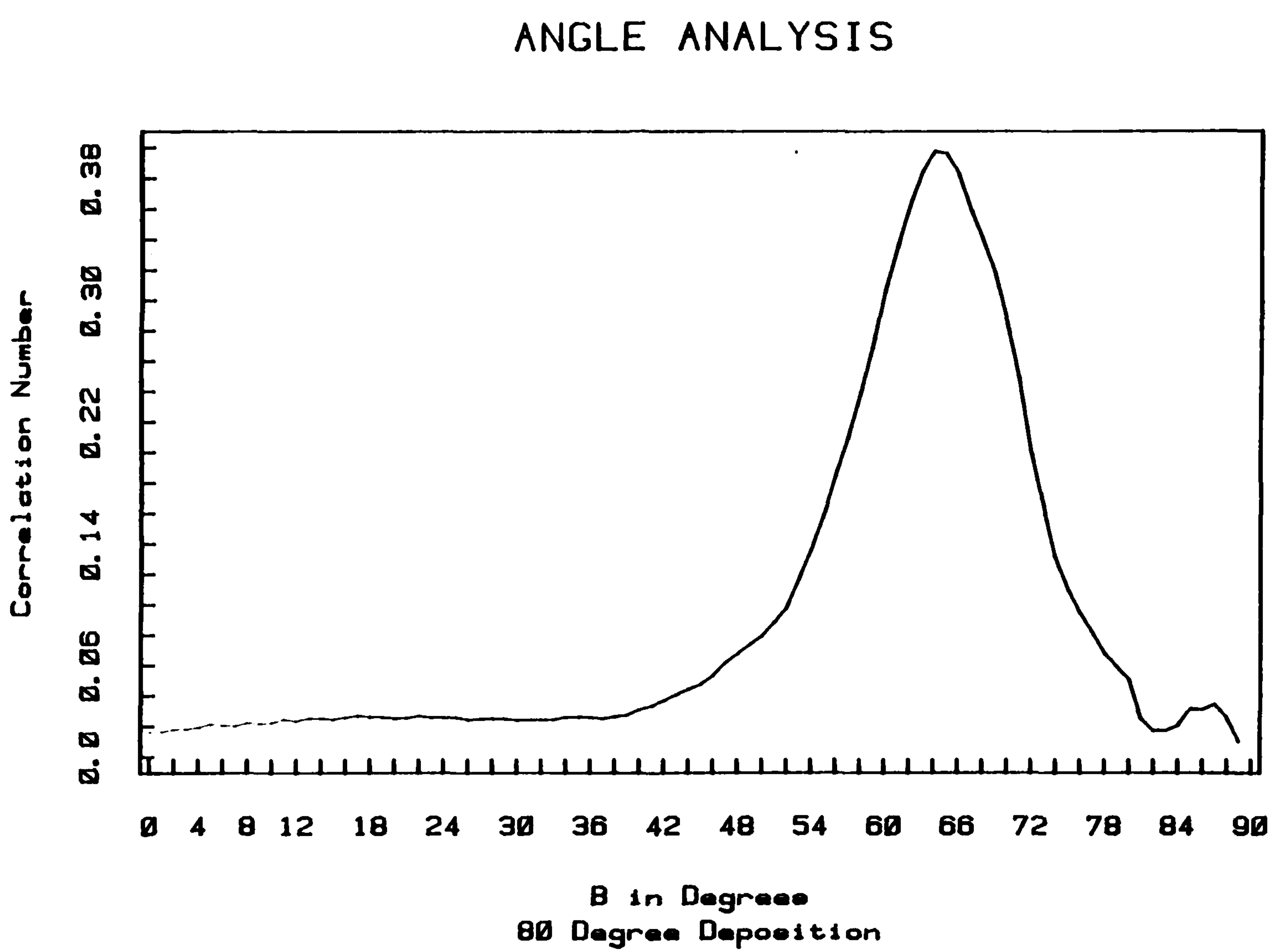


Fig. 5.14. Columnar Angle Analysis of Film Formed by 80 Degree Deposition

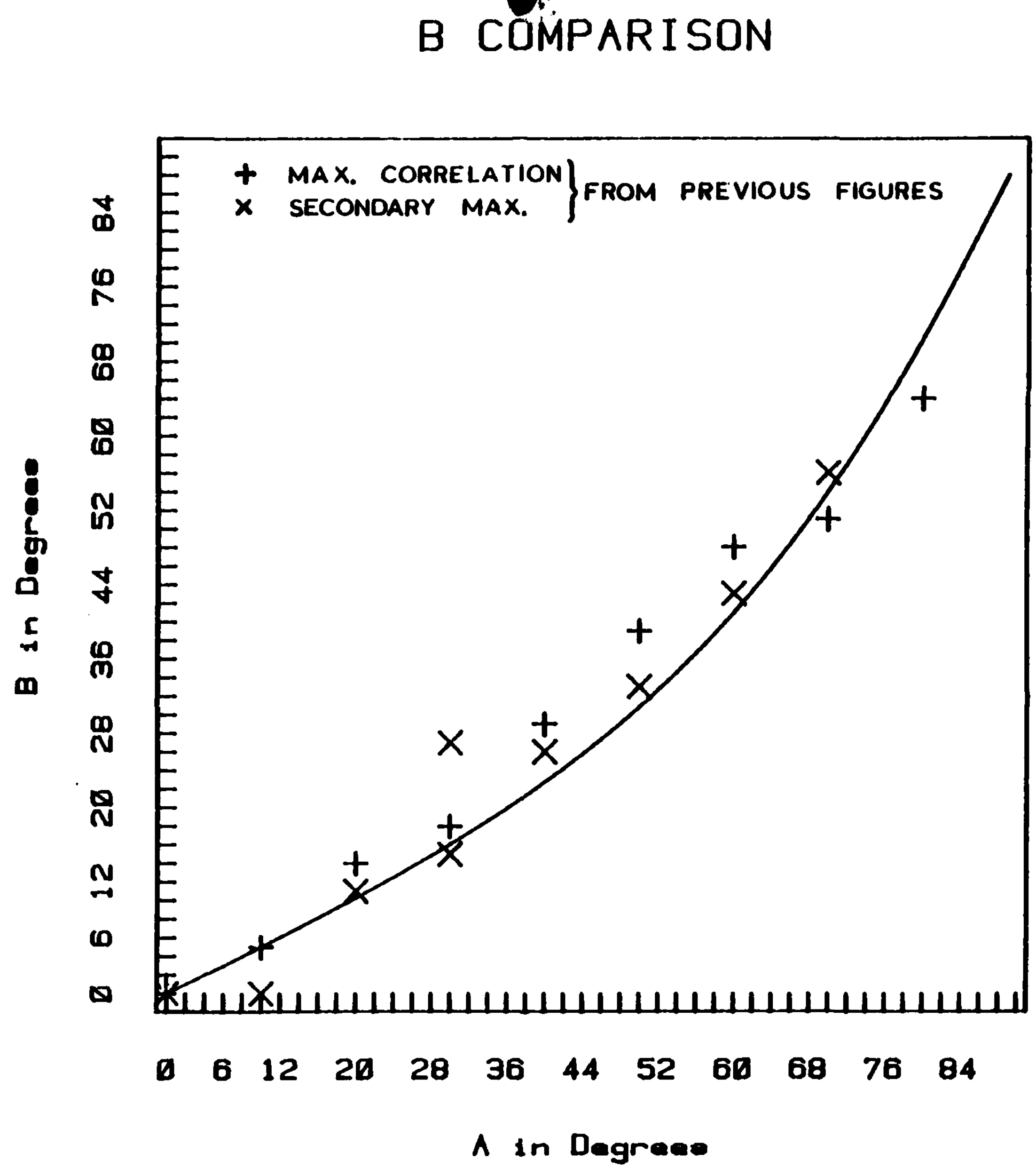


Fig. 5.15. Comparison of Column Angle Determined, With Tangent Rule Predictions

VI Conclusions and Recommendations

From the results in chapter V and the experimental results outlined in chapter II, it is evident that this simple, limited-mobility deposition scheme yields films that reproduce the general features observed experimentally. These features are:

1. Film density decreases with increasing A.
2. Column-like higher density regions appear at angle B < A. Column orientation agrees reasonably well with the tangent rule.
3. Column separation and definition increases with increasing A.

In view of the assumption made and the results obtained two logical follow on efforts could improve the computer model. First, expanding this two-dimensional model to three would reverify the results here and would also provide a means of verifying the change in columnar shape with deposition angle. Second, and most importantly, particle energy was treated in a crude manner to say the least. Consequently, the mobility of the model does not accurately portray mobilities found in real life. Since mobility and density are directly proportional, a study of a variety of model mobilities might yield some good insight into how mobility could be addressed in this model so as to obtain realistic film densities. If either or both of these model expansions take place a more accurate result should be ob-

tained. However, memory capacity plays an extremely large role in such an undertaking. The internal memory capacity required needs to be kept in a finite bound. One way of helping this situation is to reduce the size of the xy plane. Another would be to develop an algorithm which allows the internal memory to keep track of only the upper portion of the thin film. As the lower portions become buried and inactive, the data can be transferred to disk storage, thereby conserving internal memory.

As for the accuracy of the results, two comments are offered. First, density calculation for angle analysis need to be changed to allow for local density variations. Second, a great amount of weight should not be put on these column angle determinations alone. If many depositions were run for each deposition angle a more reliable determination of column angle and relative density could be obtained.

Still, the following observation can be made. Since the results reproduce the general characteristics of thin film growth, and neither momentum, oxygen absorption, crystalline texture, nor facet formation figure in the simulation process; low mobility and geometric shadowing seem to be the main driver in the development of the microstructure.

APPENDIX A

Fortran Listing of TFG Simulator

```
C
C                     "Thin Film Growth Simulator"
C                             Version 1.00
C                    Written by Jeffrey A. Stefoneck
C                               1 Oct 84
C
C
C       This program simulates vapor deposited thin film growth.
C       It was designed to run on the VAX with an HP 7220 plotter and
C       requires Fortran 77, IMSL, and S.
C
C
C
C
C               *********************************
C               *                               *
C               *     MAIN PROGRAM START-UP     *
C               *       AND INTERFACE LOOP      *
C               *                               *
C               *********************************
C
C
C
C               ********** ARRAY DEFINITION **********
C
C
00010   DOUBLE PRECISION DSEED
        REAL RN,ANG,CELCNT,SIZE
        INTEGER XAXIS,YAXIS,ZAXIS,TNDSKS
        DIMENSION DBUFF(3,240),NDSK(240)
        DIMENSION CELL(0:329,0:239,3)
CT2,3   DIMENSION PBUFFX(6),PBUFFY(6)
        DIMENSION WBUFF(100,3)
        REAL DIRIND(0:89)
        CHARACTER*2 CANS
        XAXIS=319
        YAXIS=239
        ZAXIS=3
        SIZE=0
C
C
C               ********** SIGN ON **********
C
C
00020   WRITE(6,09001)
C
```

```
C
C                *********** ARRAY INITIALIZATION ***********
C
C
        CALL INITB(DBUFF,NDSK)
        CALL INITM(CELL,TNDSKS,IHY)
        DSEED=1831728294.D0
C
C
C                *********** INTERFACE LOOP ***********
C
C
01000   WRITE(6,09010)
        READ (5,09012) DATE
        DSEED=DSEED-DATE
01001   WRITE(6,09011)
01002   READ(5,09012) CANS
        IF (CANS.EQ.'B') THEN
                GO TO 20001
        ELSE IF (CANS.EQ.'D') THEN
                GO TO 30001
        ELSE IF (CANS.EQ.'M') THEN
                GO TO 40001
CT2,3   ELSE IF (CANS.EQ.'P') THEN
CT2,3           GO TO 50001
        ELSE IF (CANS.EQ.'A') THEN
                GO TO 60001
        END IF
        WRITE(6,09013)
        GO TO 01002
C
C
C                *********** START-UP AND INTERFACE MSGS ***********
C
C
09001   FORMAT(10(2X/),25X,'Thin Film Growth(TFG) Simulator'/,34X,'Versi
     Con 1.00'/,35X,'=========='/,25X,'Written by Jeffrey A. Stefoneck'/
     C,36X,'1 Oct 84'/,9(2X/))
09010   FORMAT('Enter date (MMDDYYYY).  '$)
09011   FORMAT(7(2X/),27X,'Thin Film Growth Simulator'/,2(2x/),15X,'B-Bu
     Cffer Editor'/,15X,'D-Deposit'/,15X,'M-Matrix Manager'/,15X,'A-Anal
     Cysis'/,
CT2,3      C15X,'P-Print Cell Content'/,
     C11X,'Break-Exit to Unix'/,7(2X/))
09012   FORMAT(A1)
09013   FORMAT(15X,'Incorrect response, please try again.'/)
C
C
```

```
C
C                 ***********************************
C                 *                                 *
C                 *          BUFFER EDITOR          *
C                 *                                 *
C                 ***********************************
C
C
C
C                 ********** BUFFER EDITOR INTERFACE LOOP **********
C
C
20001   WRITE(6,29001)
20002   READ(5,29003) CANS
        IF (CANS.EQ.'S') THEN
                GO TO 21000
        ELSE IF (CANS.EQ.'M') THEN
                GO TO 22000
        ELSE IF (CANS.EQ.'A') THEN
                GO TO 23000
        ELSE IF (CANS.EQ.'R') THEN
                GO TO 24000
        ELSE IF (CANS.EQ.'D') THEN
                GO TO 26000
        ELSE IF (CANS.EQ.'E') THEN
                GO TO 01001
        ELSE IF (CANS.EQ.'ED') THEN
                GO TO 30001
        ELSE IF (CANS.EQ.'EM') THEN
                GO TO 40001
CT2,3   ELSE IF (CANS.EQ.'EP') THEN
CT2,3           GO TO 50001
        ELSE IF (CANS.EQ.'EA') THEN
                GO TO 60001
        END IF
        WRITE(6,29002)
        GO TO 20002
C
C
C                 ********** BUFFER EDITOR SHOW COMMAND **********
C
C
21000   DO 21003 L=1,20
                WRITE(6,29010)
                DO 21001 LL=1,15
                        K=15*(L-1)+LL
                        WRITE(6,29011) K,NDSK(K),(DBUFF(J,K),J=1,3)
                        IF(K.GE.240) GO TO 21002
21001           CONTINUE
21002           WRITE(6,29012)
                READ(5,29003) CANS
                IF(CANS.EQ.'E') GO TO 20001
                IF(K.GE.240) GO TO 21004
```

```
21003   CONTINUE
21004   GO TO 20001
C
C
C                ********** BUFFER EDITOR MODIFY COMMAND **********
C
C
22000   WRITE(6,29020)
22002   READ *,LENTRY
        IF ((LENTRY.GT.0).AND.(LENTRY.LE.240)) GO TO 22003
        IF (LENTRY.EQ.0) GO TO 20001
        WRITE(6,29002)
        GO TO 22002
22003   WRITE(6,29022)
22004   READ *,NDSK(LENTRY)
        S=XAXIS
        T=YAXIS
        HYP=SQRT(S**2+T**2)
        IF ((NDSK(LENTRY).GE.0).AND.(NDSK(LENTRY).LE.HYP)) GO TO 22005
        WRITE(6,29002)
        GO TO 22004
22005   IF (NDSK(LENTRY).EQ.0) GO TO 20001
        IF (NDSK(LENTRY).GT.1) GO TO 22010
        DBUFF(3,LENTRY)=0
        WRITE(6,29032)
22006   READ *,DBUFF(1,LENTRY)
        IF ((DBUFF(1,LENTRY).GT.0).AND.(DBUFF(1,LENTRY).LE.XAXIS)) GO TO
     C22007
        WRITE(6,29002)
        GO TO 22006
22007   WRITE(6,29034)
22008   READ *,DBUFF(2,LENTRY)
        IF ((DBUFF(2,LENTRY).GT.0).AND.(DBUFF(2,LENTRY).LE.YAXIS)) GO TO
     C22009
        WRITE(6,29002)
        GO TO 22008
22009   IF (NDSK(LENTRY).EQ.1) GO TO 22012
22010   DBUFF(1,LENTRY)=0
        DBUFF(2,LENTRY)=0
        WRITE(6,29035)
22011   READ *,DBUFF(3,LENTRY)
        IF ((DBUFF(3,LENTRY).GE.0).AND.(DBUFF(3,LENTRY).LT.360)) GO TO 2
     C2012
        WRITE(6,29002)
        GO TO 23011
22012   GO TO 22000
C
C
C                ********** BUFFER EDITOR ADD COMMAND **********
C
C
23000   DO 23002 L=240,1,-1
                IF (NDSK(L).GT.0) GO TO 23003
```

```
23002   CONTINUE
        LENTRY=LENTRY+1
23003   WRITE(6,29030) LENTRY
23004   READ *,NDSK(LENTRY)
        S=XAXIS
        T=YAXIS
        EMX=SQRT(S**2+T**2)
        IF ((NDSK(LENTRY).GE.0).AND.(NDSK(LENTRY).LE.EMX)) GO TO 23005
        WRITE(6,29002)
        GO TO 23004
23005   IF (NDSK(LENTRY).EQ.0) GO TO 20001
        IF (NDSK(LENTRY).GT.1) GO TO 23010
        WRITE(6,29032)
23006   READ *,DBUFF(1,LENTRY)
        IF ((DBUFF(1,LENTRY).GT.0).AND.(DBUFF(1,LENTRY).LE.XAXIS)) GO TO
     C23007
        WRITE(6,29002)
        GO TO 23006
23007   WRITE(6,29034)
23008   READ *,DBUFF(2,LENTRY)
        IF ((DBUFF(2,LENTRY).GT.0).AND.(DBUFF(2,LENTRY).LE.YAXIS)) GO TO
     C23009
        WRITE(6,29002)
        GO TO 23008
23009   IF (NDSK(LENTRY).EQ.1) GO TO 23012
23010   WRITE(6,29035)
23011   READ *,DBUFF(3,LENTRY)
        IF ((DBUFF(3,LENTRY).GE.0).AND.(DBUFF(3,LENTRY).LT.360)) GO TO 2
     C3012
        WRITE(6,29002)
        GO TO 23011
23012   LENTRY=LENTRY+1
        GO TO 23003
C
C
C                ********** BUFFER EDITOR RESET COMMAND **********
C
C
24000   CALL INITB(DBUFF,NDSK)
        GO TO 20001
C
C
C                ********** BUFFER EDITOR DELETE LINE **********
C
C
26000   WRITE(6,29060)
26002   READ *,IANS
        IF((IANS.GE.1).AND.(IANS.LE.240)) GO TO 26003
        IF(IANS.EQ.0) GO TO 20001
        WRITE(6,29061)
        GO TO 26002
26003   NDSK(IANS)=0
        DBUFF(1,IANS)=0
```

```
      DBUFF(2,IANS)=0
      DBUFF(3,IANS)=0
      GO TO 26000
C


C
C                 ********** BUFFER EDITOR MSGS **********
C
C
29001   FORMAT(6(2X/),25X,'DATA BUFFER EDITOR   ',2(2X/),15X,'S-Show Con
     Ctents of Data Buffer'/,15X,'M-Modify Contents of Data Buffer'/,15X
     C,'A-Add Data to End of Data Buffer String'/,15X,'D-Delete Data Buf
     Cfer Line'/,15X,'R-Reset Data Buffer to Zero'/,14X,'E?-Exit',6(2X/)
     C)
29002   FORMAT(2X/,15X,'Incorrect response, please try agian.',2X)
29003   FORMAT(A2,$)
C
29010   FORMAT(2X/,27('*'),' CONTENTS OF DATA BUFFER ',28('*')//,13X,'En
     Ctry N  N Disks    X-Coord        Y-Coord        Angle'/)
29011   FORMAT(15X,I3,6X,I3,6X,3(F8.4,5X))
29012   FORMAT('To scroll up, push any key (except E) and CR.'/,'To exit
     C to main menu push E and CR.')
C
29020   FORMAT('Modifying data',2X/,9X,'What line number? '$)
C29021  FORMAT(I7)
29022   FORMAT(10X,'Number of disks? '$)
C
29030   FORMAT('Adding data on line ',I3,'.'/,10X,'Number of disks
     C? '$)
C29031  FORMAT(I7)
29032   FORMAT(13X,'X Coordinate? '$)
C29033  FORMAT(F11.4)
29034   FORMAT(13X,'Y Coordinate? '$)
29035   FORMAT(20X,'Angle? '$)
C29036  FORMAT(F8.4)
C
29040   FORMAT(I3,2X,I3,2X/,I3,2X,I5)
29041   FORMAT(E15.9,2X,E15.9)
C
29060   FORMAT(2X/,'What line do you wish deleted?  '$)
29061   FORMAT(2X/,'No such line.  Answer should be 1-240.')
C
C
C
```

```
C
C                 **********************************
C                 *                                *
C                 *         TFG DEPOSITOR          *
C                 *                                *
C                 **********************************
C
C
C
C                 ********** TFG DEPOSITOR INTERFACE LOOP **********
C
C
30001   WRITE(6,39000)
30002   READ *,IANS
        IAREA=XAXIS*YAXIS*.6
        IF ((IANS.GE.0).AND.(IANS.LE.IAREA)) GO TO 30003
        IF (IANS.EQ.0) GO TO 20001
        WRITE(6,39002)
        GO TO 30002
30003   WRITE(6,39003)
30004   READ *,ANS
        IF ((ANS.GE.0).AND.(ANS.LE.80)) GO TO 30005
        WRITE(6,39005)
        GO TO 30004
30005   WRITE (6,39006) ANS,IANS
30006   READ(5,39007) CANS
        IF (CANS.EQ.'E') THEN
                GO TO 1001
        ELSE IF (CANS.EQ.'EB') THEN
                GO TO 20001
        ELSE IF (CANS.EQ.'EM') THEN
                GO TO 40001
CT2,3   ELSE IF (CANS.EQ.'EP') THEN
CT2,3           GO TO 50001
        ELSE IF (CANS.EQ.'EA') THEN
                GO TO 60001
        ELSE IF (CANS.EQ.'R') THEN
                GO TO 30001
        ELSE IF (CANS.EQ.'D') THEN
                GO TO 31001
        END IF
        WRITE(6,39008)
        GO TO 30006
C
C
C                 ********** TFG DEPOSITOR LOOP **********
C
C
31001   ANGLE=ANS
        NODSK=IANS
        RANGLE=ANGLE/57.29577951
        CA=COS(RANGLE)
        D=TAN(RANGLE)
```

```
      E=D*D+1
      C=0
      ND=0
      NDDEP=0
      DO 31002 NTDEP=NODSK,1,-1
CT3       print *,'NODSK=',NODSK
          IF (IHY.EQ.(YAXIS-1)) GO TO 31004
C
C
C                 ----- RANDOM NUMBER GENERATION -----
C
C
31100     RN=GGUBFS (DSEED)
          RAND=RN*(XAXIS+1)-1
CT3       print *,'RAND=',RAND
C
C
C                 ----- COLLISION POINT DETERMINATION -----
C
C
          Y2=0
          DO 31205 ISTEP=0,3
CT3           print *,'ISTEP=',ISTEP
              AA=RAND+1.414592654/CA
              AS=RAND+.9428090416*(ISTEP/CA)
              X=D*IHY+AS
              JUMP=0
              LX=X
              DO 31204 IYS=IHY,0,-1
CT3               print *,'IHY=',IHY,' IYS=',IYS
                  X=D*IYS+AS
                  ISHFT=(INT(X/(XAXIS+1)))*(XAXIS+1)
                  NXS=INT(X-ISHFT)
                  LXS=INT(LX-ISHFT)
                  DO 31202 IXS=LXS,0,-1
CT3                   print *,'LXS=',LXS,'IXS=',IXS,'NXS=',NXS
                      IF (CELL(IXS,IYS,3).EQ.1) THEN
                          IF ((IXS.EQ.IXT).AND.(IYS.EQ.IYT)) GO TO 312
     C07
                          XMIN=D*CELL(IXS,IYS,2)+RAND-ISHFT
                          XMAX=XMIN+2.828427125/CA
CT3                       print *,'Possible Cl (',CELL(IXS,IYS,1),',',
CT3    CCELL(IXS,IYS,2),')  XMIN=',XMIN,'  XMAX=',XMAX
                          IF (ISTEP.LT.2) THEN
                              IF ((CELL(IXS,IYS,1).LT.XMIN).OR.(CELL(I
     CXS,IYS,1).GT.XMAX)) GO TO 31207
                              A=AA-ISHFT
                              F=CELL(IXS,IYS,2)-D*(A-CELL(IXS,IYS,1))
                              G=(A-CELL(IXS,IYS,1))+D*CELL(IXS,IYS,2)
                              Y=(F+SQRT(2*E-G*G))/E
CT3                           print *,'Y coord ',Y
                              IXT=IXS
                              IYT=IYS
```

```
                                IF (Y2.LT.Y) THEN
                                    Y2=Y
                                    IX1=IXS
                                    IY1=IYS
                                    A2=A
                                END IF
                                JUMP=1
                            ELSE
                                IF ((CELL(IXS,IYS,1).LT.XMIN).OR.(CELL(I
     CXS,IYS,1).GT.XMAX)) GO TO 31207
                                A=AA-ISHFT
                                F=CELL(IXS,IYS,2)-D*(A-CELL(IXS,IYS,1))
                                G=(A-CELL(IXS,IYS,1))+D*CELL(IXS,IYS,2)
                                Y=(F+SQRT(2*E-G*G))/E
CT3                             print *,'Y Coord ',Y
                                IXT=IXS
                                IYT=IYS
                                IF (Y2.LT.Y) THEN
                                    Y2=Y
                                    IX1=IXS
                                    IY1=IYS
                                    A2=A
                                END IF
                                JUMP=1
                            END IF
                        END IF
31207                   IF (IXS.EQ.NXS) GO TO 31206
31202               CONTINUE
31206               LX=X
                    IF (JUMP.EQ.1) GO TO 31205
31204           CONTINUE
31205       CONTINUE
            X2=D*Y2+A2
            IF (IX1.LT.3) THEN
                X2=X2+XAXIS+1
                IX1=IX1+XAXIS+1
            END IF
CT3         print *,'C1 (',IX1,',',IY1,')  Y2 ',Y2
C
C
C               ----- REST POINT DETERMINATION -----
C
C
C
31300       SHORT=18
            DO 31302 J=-3,3
                IX3=J+IX1
                DO 31301 K=-3,3
                    IY3=K+IY1
                    IF ((IY3.LT.0).OR.(IYS.GT.IHY)) GO TO 31301
                    IF (CELL(IX3,IY3,3).EQ.1) THEN
CT3                     print *,'Possible Rest Cell (',IX3,',',IY3,')'
                        X31=CELL(IX3,IY3,1)-CELL(IX1,IY1,1)
```

```
                        Y31=CELL(IX3,IY3,2)-CELL(IX1,IY1,2)
                        R13SQR=(X31**2)+(Y31**2)
CT3                     print *,'R13SQR=',R13SQR
                        IF ((R13SQR.GE.1.8).AND.(R13SQR.LE.8.0)) THEN
                            R=R13SQR
                            STX=(X31**2)*(R*R)-R*((R*R)-8*(Y31**2))
                            IF (STX.LT.0) THEN
                                TX=0
                            ELSE
                                TX=SQRT(STX)
                            END IF
                            STY=(Y31**2)*(R*R)-R*((R*R)-8*(X31**2))
                            IF (STY.LT.0) THEN
                                TY=0
                            ELSE
                                TY=SQRT(STY)
                            END IF
                            DO 31303 L=-1,1,2
                                IF (X31.GE.0) THEN
                                    X2T=(X31*R+L*TX)/(2*R)
                                ELSE
                                    X2T=(X31*R-L*TX)/(2*R)
                                END IF
                                IF (Y31.LE.0) THEN
                                    Y2T=(Y31*R+L*TY)/(2*R)
                                ELSE
                                    Y2T=(Y31*R-L*TY)/(2*R)
                                END IF
                                X22=X2T+CELL(IX1,IY1,1)-X2
                                Y22=Y2T+CELL(IX1,IY1,2)-Y2
                                R22SQR=(X22**2)+(Y22**2)
CT3                             print *,'R22SQR=',R22SQR
                                IF (R22SQR.LT.SHORT) THEN
                                    SHORT=R22SQR
                                    X2F=X2T
                                    Y2F=Y2T
                                    IX3F=IX3
                                    IY3F=IY3
                                    FX31=X31
                                    FY31=Y31
                                    FR13SQ=R13SQR
                                END IF
31303                       CONTINUE
                        END IF
                    END IF
31301           CONTINUE
31302       CONTINUE
            R13=SQRT(FR13SQ)
CT3         print *,'R13=',R13
            X21=X2F
            Y21=Y2F
            R12SQR=X21**2+Y21**2
            X2=X2F+CELL(IX1,IY1,1)
```

```
             Y2=Y2F+CELL(IX1,IY1,2)
             IX2=INT(X2)
             IY2=INT(Y2)
CT3          X23=X2-CELL(IX3F,IY3F,1)
CT3          Y23=Y2-CELL(IX3F,IY3F,2)
CT3          R23SQR=X23**2+Y23**2
CT3          IF (((R12SQR.LE.1.8).OR.(R12SQR.GE.2.2)).OR.((R23SQR.LE.1.8)
CT3      C.OR.(R23SQR.GE.2.2))) THEN
CT3              print *,'Rest Point Error--R12SQR=',R12SQR,'R23SQR=',R23
CT3      CSQR
CT3              GO TO 80000
CT3          END IF
CT3          IF (CELL(IX2,IY2,3).EQ.1) THEN
CT3              print *,'Overwrite on cell (',IX2,',',IY2,')'
CT3              GO TO 80000
CT3          END IF
             IF (IX2.GT.XAXIS) THEN
                 X2M=X2-(XAXIS+1)
                 IX2M=IX2-(XAXIS+1)
                 CELL(IX2M,IY2,1)=X2M
                 CELL(IX2M,IY2,2)=Y2
                 CELL(IX2M,IY2,3)=1
             ELSE IF (IX2.LT.10) THEN
                 X2M=X2+(XAXIS+1)
                 IX2M=IX2+(XAXIS+1)
                 CELL(IX2M,IY2,1)=X2M
                 CELL(IX2M,IY2,2)=Y2
                 CELL(IX2M,IY2,3)=1
             END IF
             CELL(IX2,IY2,1)=X2
             CELL(IX2,IY2,2)=Y2
             CELL(IX2,IY2,3)=1
CT3          print *,'Final position (',X2,',',Y2,')'
             TNDSKS=TNDSKS+1
             NDDEP=NDDEP+1
             ND=ND+1
             IF (ND.EQ.50) THEN
                 WRITE(6,39010) NDDEP
                 ND=0
             END IF
             IF (IHY.LT.(Y2+1)) THEN
                 IHY=Y2+1
             END IF
31002    CONTINUE
31003    WRITE(6,39011) NDDEP
         GO TO 01001
31004    WRITE(6,39012) NDDEP
         GO TO 01001
C
```

```
C
C                  *********** TFG DEPOSITOR MESSAGES **********
C
C
39000   FORMAT(2X/,'Number of disks to be deposited?  '$)
C39001  FORMAT(I5)
39002   FORMAT(2X/,'You are depositing more disks than there are cells i
    Cn matrix!'/,'Please try again, but keep it under matrix max.')
39003   FORMAT(2X/,'What angle do you want it deposited at?  '$)
C39004  FORMAT(F7.4)
39005   FORMAT(2X/,'Angle to large.  Please try again.')
39006   FORMAT(10(2X/),'Deposition will occur at ',F8.4,' degrees, and w
    Cill deposit ',I5,' disks max.'/,4(2X/),10X,'D-Deposit as specified
    C'/,10X,'R-Reenter angle disk number'/,10X,'E?-Exit',5(2X/))
39007   FORMAT(A2)
39008   FORMAT(2X/,'Incorrect response.  Please try again.')
39010   FORMAT(I5,' disks deposited.')
39011   FORMAT(I5,' disks deposited--DEPOSITON COMPLETED')
39012   FORMAT('Spatail matrix full, deposition halted with ',I5,' disks
    C deposited.')
C
C
C
```

```
C
C                   **********************************
C                   *                                *
C                   *         MATRIX MANAGER         *
C                   *                                *
C                   **********************************
C
C
C
C                   ********** MATRIX MANAGER INTERFACE LOOP **********
C
C
40001   WRITE(6,49000)
40002   READ(5,49002) CANS
        FLAG0=1
        IF (CANS.EQ.'RB') THEN
                GO TO 41000
        ELSE IF (CANS.EQ.'R') THEN
                GO TO 45000
        ELSE IF (CANS.EQ.'MA') THEN
                GO TO 41000
        ELSE IF (CANS.EQ.'MD') THEN
                GO TO 42001
        ELSE IF (CANS.EQ.'W') THEN
                GO TO 43000
        ELSE IF (CANS.EQ.'E') THEN
                GO TO 01001
        ELSE IF (CANS.EQ.'EB') THEN
                GO TO 20001
        ELSE IF (CANS.EQ.'ED') THEN
                GO TO 30001
CT2,3   ELSE IF (CANS.EQ.'EP') THEN
CT2,3           GO TO 50001
        ELSE IF (CANS.EQ.'EA') THEN
                GO TO 60001
        ELSE IF (CANS.EQ.'MC') THEN
                GO TO 44000
        ELSE IF (CANS.EQ.'WS') THEN
                GO TO 46000
        ELSE IF (CANS.EQ.'RS') THEN
                GO TO 47000
        ELSE IF (CANS.EQ.'SA') THEN
                GO TO 48100
        ELSE IF (CANS.EQ.'SB') THEN
                GO TO 48200
        ELSE IF (CANS.EQ.'SC') THEN
                GO TO 48300
        END IF
        WRITE(6,49001)
        GO TO 40002
C
```

```
C
C              ********** MATRIX READ BUFFER/ADD COMMAND **********
C
C
41000   FLAG0=0
C
C
C              ********** MATRIX DELETE COMMAND **********
C
C
42001   PAF=1
        IF (CANS.EQ.'RB') THEN
                TNDSKS=0
                IHY=0
        END IF
        DO 42009 LENTRY=1,240
                IF (NDSK(LENTRY).EQ.0) GO TO 42009
                IF (NDSK(LENTRY).GT.1) GO TO 42003
                IX=INT(DBUFF(1,LENTRY))
                IY=INT(DBUFF(2,LENTRY))
                IF (CELL(IX,IY,3).EQ.FLAG0) GO TO 42002
                WRITE(6,49020) LENTRY
                GO TO 40001
42002           IF (FLAG0.EQ.0) THEN
                        CELL(IX,IY,1)=DBUFF(1,LENTRY)
                        CELL(IX,IY,2)=DBUFF(2,LENTRY)
                        CELL(IX,IY,3)=1
                        IF (IX.LT.10) THEN
                                XM=DBUFF(1,LENTRY)+XAXIS+1
                                IXM=INT(XM)
                                CELL(IXM,IY,1)=XM
                                CELL(IXM,IY,2)=DBUFF(2,LENTRY)
                                CELL(IXM,IY,3)=1
                        END IF
                        TNDSKS=TNDSKS+1
                        IF (IHY.LT.IY) THEN
                                IHY=IY
                        END IF
                ELSE
                        CELL(IX,IY,1)=0
                        CELL(IX,IY,1)=0
                        CELL(IX,IY,3)=0
                        IF (IX.LT.10) THEN
                                XM=DBUFF(1,LENTRY)+XAXIS+1
                                IXM=INT(XM)
                                CELL(IXM,IY,1)=0
                                CELL(IXM,IY,2)=0
                                CELL(IXM,IY,3)=0
                        END IF
                        TNDSKS=TNDSKS-1
                        IF (IHY.GT.IY) THEN
                                IHY=IY
                        END IF
```

```
              END IF
              PX=CELL(IX,IY,1)
              PY=CELL(IX,IY,2)
CT2           print *,'Point entry at (',PX,',',PY,')'
              PAF=0
              GO TO 42009
42003         RANGLE=DBUFF(3,LENTRY)/57.29577951
              IF (PAF.LT.1) GO TO 42004
              PAH=PA+1.047197551
              PAL=PA-1.047197551
              IF (PAH.GE.6.283185307) THEN
                    PAH=PAH-6.283185307
              END IF
              IF (PAL.LT.0) THEN
                    PAL=PAL+6.283185307
              END IF
              IF ((RANGLE.GE.PAH).OR.(RANGLE.LE.PAL)) GO TO 42004
              WRITE(6,49021) LENTRY
              GO TO 40001
42004         DO 42007 LSTEP=1,400
                    X=PX+(LSTEP*1.414213562*COS(RANGLE))
                    Y=PY+(LSTEP*1.414213562*SIN(RANGLE))
                    IF (((X.GE.0).AND.(X.LE.XAXIS)).AND.((Y.GE.0).AN
     CD.(Y.LE.YAXIS))) GO TO 42005
                    WRITE(6,49022) LENTRY
                    GO TO 40001
42005               IX=INT(X)
                    IY=INT(Y)
                    IF (CELL(IX,IY,3).EQ.FLAG0) GO TO 42006
                    WRITE(6,49020) LENTRY
                    GO TO 40001
42006               IF (FLAG0.EQ.0) THEN
                          CELL(IX,IY,1)=X
                          CELL(IX,IY,2)=Y
                          CELL(IX,IY,3)=1
                          IF (IX.LT.10) THEN
                                XM=X+XAXIS+1
                                IXM=INT(XM)
                                CELL(IXM,IY,1)=XM
                                CELL(IXM,IY,2)=Y
                                CELL(IXM,IY,3)=1
                          END IF
                          TNDSKS=TNDSKS+1
                          IF (IHY.LT.IY) THEN
                                IHY=IY
                          END IF
                    ELSE
                          CELL(IX,IY,1)=0
                          CELL(IX,IY,2)=0
                          CELL(IX,IY,3)=0
                          IF (IX.LT.10) THEN
                                XM=X+XAXIS+1
                                IXM=INT(XM)
```

```
                                        CELL(IXM,IY,1)=0
                                        CELL(IXM,IY,2)=0
                                        CELL(IXM,IY,3)=0
                                END IF
                                TNDSKS=TNDSKS-1
                                IF (IHY.GT.IY) THEN
                                        IHY=IY
                                END IF
                        END IF
                        IF (NDSK(LENTRY).EQ.LSTEP) GO TO 42008
42007           CONTINUE
42008           PX=X
                PY=Y
CT2             print *,'Line entry ending at (',PX,',',PY,')'
                PAF=1
                IF (RANGLE.LE.3.141592654) THEN
                        PA=RANGLE+3.141592654
                ELSE
                        PA=RANGLE-3.141592654
                END IF
42009   CONTINUE
        GO TO 40001
C
C
C               ********** MATRIX WRITE FILE COMMAND **********
C
C
43000   OPEN(3,FILE='dep',STATUS='NEW')
        REWIND(3)
        WRITE(3,49030) XAXIS,YAXIS,IHY,TNDSKS
        CLOSE(3)
        OPEN(3,FILE='plot',STATUS='NEW')
        REWIND(3)
43003   L=1
        DO 43005 J=XAXIS,0,-1
            DO 43004 K=YAXIS,0,-1
                IF (CELL(J,K,3).EQ.1) THEN
                    WBUFF(L,1)=CELL(J,K,1)
                    WBUFF(L,2)=CELL(J,K,2)
                    WBUFF(L,3)=SIZE
                    L=L+1
                    IF (L.EQ.6) THEN
                        WRITE(3,49033) ((WBUFF(M,N),N=1,3),M=5,1,-1)
                        L=1
                    END IF
                END IF
                IF ((J.EQ.0).AND.(K.EQ.O)) THEN
                    L=L-1
                    WRITE(3,49033) ((WBUFF(M,N),N=1,3),M=L,1,-1)
                END IF
43004       CONTINUE
43005   CONTINUE
        CLOSE(3)
```

```
      GO TO 40001
C




C
C               ********** MATRIX CLEAR COMMAND **********
C
C
44000 CALL INITM(CELL,TNDSKS,IHY)
      GO TO 40001
C
C
C               ********** MATRIX READ FILE COMMAND **********
C
C
45000 OPEN(2,FILE='dep',STATUS='OLD')
      REWIND(2)
      READ(2,49030) XAXIS,YAXIS,IHY,TNDSKS
      CLOSE(2)
      OPEN(2,FILE='plot',STATUS='OLD')
      REWIND(2)
45001 DO 45002 J=TNDSKS,1,-5
              IF (J.LE.5) THEN
                      K=J
              ELSE
                      K=5
              END IF
              READ(2,49033) ((WBUFF(M,N),N=1,3),M=K,1,-1)
              DO 45003 N=K,1,-1
                      IX=INT(WBUFF(N,1))
                      IY=INT(WBUFF(N,2))
                      CELL(IX,IY,1)=WBUFF(N,1)
                      CELL(IX,IY,2)=WBUFF(N,2)
                      CELL(IX,IY,3)=1
                      IF (WBUFF(N,1).LT.10) THEN
                              XM=WBUFF(N,1)+XAXIS+1
                              IXM=INT(XM)
                              CELL(IXM,IY,1)=XM
                              CELL(IXM,IY,2)=WBUFF(N,2)
                              CELL(IXM,IY,3)=1
                      END IF
45003         CONTINUE
45002 CONTINUE
      CLOSE(2)
      SIZE=WBUFF(N,3)
      GO TO 40001
C
```

```
C
C                  ********** MATRIX WRITE SUBS FILE COMMAND **********
C
C
46000    OPEN(3,FILE='subd',STATUS='NEW')
         REWIND(3)
         WRITE(3,49030) XAXIS,YAXIS,IHY,TNDSKS
         CLOSE(3)
         OPEN(3,FILE='subp',STATUS='NEW')
         REWIND(3)
         GO TO 43003
C
C
C                  ********** MATRIX READ SUBS FILE COMMAND **********
C
C
47000    OPEN(2,FILE='subd',STATUS='OLD')
         REWIND(2)
         READ(2,49030) XAXIS,YAXIS,IHY,TNDSKS
         CLOSE(2)
         OPEN(2,FILE='subp',STATUS='OLD')
         REWIND(2)
         GO TO 45001
C
C
C                  ********** MATRIX SET AXES COMMAND **********
C
C
48100    CALL INITM(CELL,TNDSKS,IHY)
         XAXIS=319
         YAXIS=239
         SIZE=0
         GO TO 40001
C
48200    CALL INITM(CELL,TNDSKS,IHY)
         XAXIS=159
         YAXIS=119
         SIZE=.3
         GO TO 40001
C
48300    CALL INITM(CELL,TNDSKS,IHY)
         XAXIS=79
         YAXIS=59
         SIZE=.7
         GO TO 40001
C
C
C                  ********** MATRIX MANAGER MESSAGES **********
C
C
49000    FORMAT(2(2X/),25X,'MATRIX MANAGER',2(2X/),10X,'Set Axes'/,15X,'S
     CA   320X240'/,15X,'SB   160X120'/,15X,'SC    80X60'/,10X,'Read Dat
     Ca Into Matrix From'/,15X,'R    Plot and Dep Files'/,15X,'RB   Buff
```

```
     Cer'/,15X,'RS   Subp and Subd Files '/,10X,'Write from Matrix to'/,
     C15X,'W    Plot and Dep Files'/,15X,'WS   Subp and Subd Files'/,10X
     C,'Modify Matrix'/,15X,'MA   Add Buffer Data to Matrix'/,15X,'MC
     CClear Matrix'/,15X,'MD   Delete Buffer Data from Matrix'///,15X,'E
     C?   Exit',2(2X/))
49001   FORMAT(2X/,15X,'Incorrect response, please try agian.',2X)
49002   FORMAT(A2,$)
C
49020   FORMAT('Cell required by line ',I3,' is already filled/delet
     Ced.  File aborted.')
49021   FORMAT('Angle required by line ',I3,' is to sharp.  File abo
     Crted.')
49022   FORMAT('Surface specified by line ',I3,' is out of matrix.
     CFile aborted.')
C
49030   FORMAT(I3,2X,I3,2X/,I3,2X,I5)
49033   FORMAT(5(F6.2,1X,F6.2,1X,F2.1,1X),:)
49034   FORMAT(5(F2.1,1X,F6.2,1X,F6.2,1X),:)
C
C
C
```

```
C
C                 **********************************
C                 *                                *
C                 *       PRINT CELL CONTENT       *
C                 *                                *
C                 **********************************
C
C
C
C                 ********** PRINT CELL CONTENT LOOP **********
C
C
CT2,350001        PRINT *,'PRINT CELL CONTENT'
CT2,3     L=0
CT2,3     DO 50003 J=XAXIS,0,-1
CT2,3             DO 50002 K=YAXIS,0,-1
CT2,3                     IF (CELL(J,K,3).EQ.1) THEN
CT2,3                             L=L+1
CT2,3                             PBUFFX(L)=CELL(J,K,1)
CT2,3                             PBUFFY(L)=CELL(J,K,2)
CT2,3                     END IF
CT2,3                     IF ((J.EQ.0).AND.(K.EQ.0)) THEN
CT2,3                             WRITE(6,59001) (PBUFFX(M),M=L,1,-1)
CT2,3                             WRITE(6,59001) (PBUFFY(M),M=L,1,-1)
CT2,3                     ELSE IF (L.EQ.6) THEN
CT2,3                             WRITE(6,59000) (PBUFFX(M),M=L,1,-1),(PBU
CT2,3       CFFY(N),N=L,1,-1)
CT2,3                             L=0
CT2,3                     END IF
CT2,350002                CONTINUE
CT2,350003        CONTINUE
CT2,350004        GO TO 01001
C
C
C                 ********** PRINT CELL CONTENT MESSAGES **********
C
C
CT2,359000        FORMAT(6(F8.4,2X)/,6(F8.4,2X)/)
CT2,359001        FORMAT(6(F8.4,2X))
C
```

```
C
C                  ***********************************
C                  *                                 *
C                  *            ANALYSIS             *
C                  *                                 *
C                  ***********************************
C
C
C
C                  ********** ANALYSIS INTERFACE LOOP
C
C
60001   WRITE(6,69000)
60002   READ(5,69001) CANS
        IF (CANS.EQ.'D') THEN
                GO TO 60003
        ELSE IF (CANS.EQ.'A') THEN
                GO TO 60003
        ELSE IF (CANS.EQ.'E') THEN
                GO TO 01001
        ELSE IF (CANS.EQ.'EB') THEN
                GO TO 20001
        ELSE IF (CANS.EQ.'ED') THEN
                GO TO 30001
        ELSE IF (CANS.EQ.'EM') THEN
                GO TO 40001
CT2,3   ELSE IF (CANS.EQ.'EP') THEN
CT2,3           GO TO 50001
        END IF
        WRITE(6,69002)
        GO TO 60002
60003   WRITE(6,69003) XAXIS,YAXIS,IHY
60004   WRITE(6,69004)
        READ(5,69005) IANS1
        WRITE(6,69006)
        READ(5,69005) IANS2
        IF ((IANS1.GT.IANS2).AND.(IANS1.LT.(YAXIS+1)).AND.(IANS2.GE.0))
     CGO TO 61000
        WRITE(6,69007)
        GO TO 60004
C
C
C                  ********** DENSITY ANALYSIS LOOP **********
C
C
61000   DCNT=0
        CELCNT=0
        DO 61002 K=IANS1,0,-1
            DO 61001 J=XAXIS,0,-1
                IF (CELL(J,K,3).EQ.1) THEN
                    DCNT=DCNT+1
                END IF
                CELCNT=CELCNT+1
```

```
61001        CONTINUE
             IF (K.EQ.IANS2) GO TO 61003
61002    CONTINUE
61003    DEN=DCNT/CELCNT
         WRITE(6,69010) DEN
         RDEN=DEN*1.732050808
         IF (CANS.EQ.'D') GO TO 60001
C
C
C                ********** ANGLE ANALYSIS LOOP **********
C
C
62000    IUY=IANS1
         ILY=IANS2
         MLEN=XAXIS+IUY-ILY-1
         DO 62013 M=0,89
             DIRIND(M)=0
62013    CONTINUE
         DO 62009 J=0,89
             ANG=J
             WRITE(6,69022) ANG
             RANG=ANG*.0174532925
             D=TAN(RANG)
             PNTR=0
             DO 62011 K=0,319
                 IF (K.GE.XAXIS) GO TO 62011
CT4              print *,'K =',K
                 DSKCNT=0
                 CELCNT=0
                 X=D*IUY+K
                 LX=X
                 DO 62008 IYS=IUY,0,-1
CT4                  print *,'IUY=',IUY,'IYS=',IYS
                     X=D*IYS+K
                     ISHFT=(INT(X/(XAXIS+1)))*(XAXIS+1)
                     NXS=INT(X-ISHFT)
                     LXS=INT(LX-ISHFT)
                     DO 62007 IXS=LXS,0,-1
CT4                      print *,'LXS=',LXS,'IXS=',IXS,'NXS=',NXS
                         IF (CELL(IXS,IYS,3).EQ.1) THEN
                             DSKCNT=DSKCNT+1
CT4                          print *,'DSKCNT =',DSKCNT
                         END IF
                         IXS2=IXS+1
                         IF (CELL(IXS2,IYS,3).EQ.1) THEN
                             DSKCNT=DSKCNT+1
                         END IF
                         CELCNT=CELCNT+2
CT4                      print *,'CELCNT =',CELCNT
                         IF (CELCNT.GE.MLEN) GO TO 62012
                         IF (IXS.EQ.NXS) GO TO 62006
62007                CONTINUE
62006                LX=X
```

```
                       IF (IYS.LE.ILY) GO TO 62012
62008              CONTINUE
62012              DENLIN=((DSKCNT*1.732050808)/CELCNT)
                   IF (DENLIN.LE.RDEN) THEN
                       S=((RDEN-DENLIN)/RDEN)**2
                   ELSE
                       S=((DENLIN-RDEN)/(1-RDEN))**2
                   END IF
CT4                print *,'S =',S
                   PNTR=PNTR+S
CT4                print *,'PNTR =',PNTR
62011          CONTINUE
62014          DIRIND(J)=PNTR/(XAXIS+1)
62009      CONTINUE
62010      print *,'COMPLETE'
           OPEN(3,FILE='ans',STATUS='NEW')
           REWIND(3)
           WRITE(3,69020) (DIRIND(N),N=0,89)
           WRITE(3,69021) DEN,RDEN
           CLOSE(3)
           GO TO 60001
C
C
C                  ********** ANALYSIS MESSAGES **********
C
C
69000      FORMAT(7(2X/),32X,'ANALYSIS PROGRAM'///,15X,'D-Density'/,15X,'A=
     CAngle'/,14X,'E?=Exit',7(2X/))
69001      FORMAT(A2)
69002      FORMAT(15X,'Incorrect response, please try again.'/)
69003      FORMAT(12(2X/),32X,'ANALYSIS PROGRAM'///,5X,'Current matrix dime
     Cnsions are ',I3,'X',I3,'.'//,5X,'Current deposition height is ',I3
     C,'.'////)
69004      FORMAT(5X,'What do you want the analysis upper bound to be? '$)
69005      FORMAT(I3)
69006      FORMAT(/5X,'What do you want the analysis lower bound to be? '$)
69007      FORMAT(/5X,'Improper boundary conditions.  Please try again.')
69010      FORMAT('Density is ',F7.4,'disks per square unit')
69020      FORMAT(9(10(F7.6,1X),/))
69021      FORMAT(F12.8,1X,F12.8)
69022      FORMAT('Angle ',F4.1)
C
```

```
C
C              ********************************
C              *                              *
C              *          ERROR TRAP          *
C              *                              *
C              ********************************
C
C
C
C
C              ********** DEPOSITION LOOP **********
C
C
CT380000               print *,'RAND=',RAND
C
C
C              ----- COLLISION POINT DETERMINATION -----
C
C
CT3          Y2=0
CT3          DO 81205 ISTEP=0,3
CT3              print *,'ISTEP=',ISTEP
CT3              AA=RAND+1.414592654/CA
CT3              AS=RAND+.9428090416*(ISTEP/CA)
CT3              X=D*IHY+AS
CT3              JUMP=0
CT3              LX=X
CT3              DO 81204 IYS=IHY,0,-1
CT3                  print *,'IHY=',IHY,' IYS=',IYS
CT3                  X=D*IYS+AS
CT3                  ISHFT=(INT(X/(XAXIS+1)))*(XAXIS+1)
CT3                  NXS=INT(X-ISHFT)
CT3                  LXS=INT(LX-ISHFT)
CT3                  DO 81202 IXS=LXS,0,-1
CT3                      print *,'LXS=',LXS,'IXS=',IXS,'NXS=',NXS
CT3                      IF (CELL(IXS,IYS,3).EQ.1) THEN
CT3                          IF ((IXS.EQ.IXT).AND.(IYS.EQ.IYT)) GO TO 812
CT3      C07
CT3                          XMIN=D*CELL(IXS,IYS,2)+RAND-ISHFT
CT3                          XMAX=XMIN+2.828427125/CA
CT3                          print *,'Possible C1 (',CELL(IXS,IYS,1),',',
CT3      CCELL(IXS,IYS,2),')  XMIN=',XMIN,'  XMAX=',XMAX
CT3                          IF (ISTEP.LT.2) THEN
CT3                              IF ((CELL(IXS,IYS,1).LT.XMIN).OR.(CELL(I
CT3      CXS,IYS,1).GT.XMAX)) GO TO 81207
CT3                              A=AA-ISHFT
CT3                              F=CELL(IXS,IYS,2)-D*(A-CELL(IXS,IYS,1))
CT3                              G=(A-CELL(IXS,IYS,1))+D*CELL(IXS,IYS,2)
CT3                              Y=(F+SQRT(2*E-G*G))/E
CT3                              print *,'Y coord ',Y
CT3                              IXT=IXS
CT3                              IYT=IYS
CT3                              IF (Y2.LT.Y) THEN
```

```
CT3                                     Y2=Y
CT3                                     IX1=IXS
CT3                                     IY1=IYS
CT3                                     A2=A
CT3                                 END IF
CT3                                 JUMP=1
CT3                             ELSE
CT3                                 IF ((CELL(IXS,IYS,1).LT.XMIN).OR.(CELL(I
CT3      CXS,IYS,1).GT.XMAX)) GO TO 81207
CT3                                 A=AA-ISHFT
CT3                                 F=CELL(IXS,IYS,2)-D*(A-CELL(IXS,IYS,1))
CT3                                 G=(A-CELL(IXS,IYS,1))+D*CELL(IXS,IYS,2)
CT3                                 Y=(F+SQRT(2*E-G*G))/E
CT3                                 print *,'Y Coord ',Y
CT3                                 IXT=IXS
CT3                                 IYT=IYS
CT3                                 IF (Y2.LT.Y) THEN
CT3                                     Y2=Y
CT3                                     IX1=IXS
CT3                                     IY1=IYS
CT3                                     A2=A
CT3                                 END IF
CT3                                 JUMP=1
CT3                             END IF
CT3                         END IF
CT381207                            IF (IXS.EQ.NXS) GO TO 81206
CT381202                        CONTINUE
CT381206                        LX=X
CT3                     IF (JUMP.EQ.1) GO TO 81205
CT381204                    CONTINUE
CT381205          CONTINUE
CT3            X2=D*Y2+A2
CT3            print *,'X2 before shift=',X2
CT3            IF (IX1.LT.3) THEN
CT3                X2=X2+XAXIS+1
CT3                IX1=IX1+XAXIS+1
CT3            END IF
CT3            print *,'C1 (',IX1,',',IY1,')  X2=',X2,'Y2=',Y2
C
C
C                ----- REST POINT DETERMINATION -----
C
C
C
CT381300            SHORT=18
CT3            DO 81302 J=-3,3
CT3                IX3=J+IX1
CT3                DO 81301 K=-3,3
CT3                    IY3=K+IY1
CT3                    IF ((IY3.LT.0).OR.(IYS.GT.IHY)) GO TO 81301
CT3                    IF (CELL(IX3,IY3,3).EQ.1) THEN
CT3                        print *,'Possible Rest Cell (',IX3,',',IY3,')'
CT3                        X31=CELL(IX3,IY3,1)-CELL(IX1,IY1,1)
```

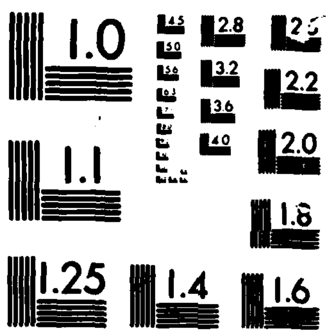
```
CT3                    Y31=CELL(IX3,IY3,2)-CELL(IX1,IY1,2)
CT3                    R13SQR=(X31**2)+(Y31**2)
CT3                    print *,CELL(IX3,IY3,1),CELL(IX1,IY1,1),CELL(
CT3     CIX3,IY3,2),CELL(IX1,IY1,2)
CT3                    print *,'R13SQR=',R13SQR
CT3                    IF ((R13SQR.GE.1.8).AND.(R13SQR.LE.8.0)) THEN
CT3                        R=R13SQR
CT3                        STX=(X31**2)*(R*R)-R*((R*R)-8*(Y31**2))
CT3                        IF (STX.LT.0) THEN
CT3                            TX=0
CT3                        ELSE
CT3                            TX=SQRT(STX)
CT3                        END IF
CT3                        STY=(Y31**2)*(R*R)-R*((R*R)-8*(X31**2))
CT3                        IF (STY.LT.0) THEN
CT3                            TY=0
CT3                        ELSE
CT3                            TY=SQRT(STY)
CT3                        END IF
CT3                        DO 81303 L=-1,1,2
CT3                            IF (X31.GE.0) THEN
CT3                                X2T=(X31*R+L*TX)/(2*R)
CT3                            ELSE
CT3                                X2T=(X31*R-L*TX)/(2*R)
CT3                            END IF
CT3                            IF (Y31.LE.0) THEN
CT3                                Y2T=(Y31*R+L*TY)/(2*R)
CT3                            ELSE
CT3                                Y2T=(Y31*R-L*TY)/(2*R)
CT3                            END IF
CT3                            X22=X2T+CELL(IX1,IY1,1)-X2
CT3                            Y22=Y2T+CELL(IX1,IY1,2)-Y2
CT3                            R22SQR=(X22**2)+(Y22**2)
CT3                            print *,'R22SQR=',R22SQR
CT3                            IF (R22SQR.LT.SHORT) THEN
CT3                                SHORT=R22SQR
CT3                                print *,'SHORT=',SHORT
CT3                                X2F=X2T
CT3                                Y2F=Y2T
CT3                                IX3F=IX3
CT3                                IY3F=IY3
CT3                                FX31=X31
CT3                                FY31=Y31
CT3                                FR13SQ=R13SQR
CT3                            END IF
CT381303                           CONTINUE
CT3                    END IF
CT3                END IF
CT381301               CONTINUE
CT381302           CONTINUE
CT3         R13=SQRT(FR13SQ)
CT3         print *,'R13=',R13
CT3         X21=X2F
```

```
CT3          Y21=Y2F
CT3          R12SQR=X21**2+Y21**2
CT3          X2=X2F+CELL(IX1,IY1,1)
CT3          Y2=Y2F+CELL(IX1,IY1,2)
CT3          print *,'X2=',X2,'Y2=',Y2
CT3          IX2=INT(X2)
CT3          IY2=INT(Y2)
CT3          X23=X2-CELL(IX3F,IY3F,1)
CT3          Y23=Y2-CELL(IX3F,IY3F,2)
CT3          R23SQR=X23**2+Y23**2
CT3          IF (IX2.GT.XAXIS) THEN
CT3              X2=X2-(XAXIS+1)
CT3          END IF
CT3          print *,'Final position (',X2,',',Y2,')'
CT3          print *,'OVERWRITE ERROR--PROGRAM HALTED(',NDDEP,')'
99999     END
C
C
```

```
C
C                ********************************
C                *                              *
C                *      SYSTEM SUBROUTINES      *
C                *                              *
C                ********************************
C
C
C
        SUBROUTINE INITB(DBUFF,NDSK)
        INTEGER NDSK(240)
        REAL DBUFF(3,240)
00030   DO 00032 L=1,3
                DO 00031 LL=1,240
                        DBUFF(L,LL)=0
00031           CONTINUE
00032   CONTINUE
        DO 00033 L=1,240
                NDSK(L)=0
00033   CONTINUE
        END
C
C
        SUBROUTINE INITM(CELL,TNDSKS,IHY)
        REAL CELL(0:329,0:239,3)
00034   DO 00037 L=0,329
                DO 00036 LL=0,239
                        CELL(L,LL,3)=0
00036           CONTINUE
00037   CONTINUE
        TNDSKS=0
        IHY=0
        END
C
C
```

AD-A167 152 COMPUTER MODELING OF THIN FILM GROWTH(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING J A STEFONECK DEC 84 AFIT/GEP/PH/84D-10 2/2

UNCLASSIFIED F/G 7/4 NL



MICROCOP' CHART

Appendix B

Derivation

Incident Disk Coordinates After Collision

The incident disk coordinates after collision are used by the collision point determination algorythm discussed in Section V. The formulas for these coordinates are used in the TFG Simulator and are derived here (See Appendix A for the coded routine and Figure 3.10 for a diagram).

Y Coordinate. The equation of a line in the slope interecept form is:

$$X = \frac{1}{m}Y + a$$

where

m = slope
a = x intercept

Also, the equation for a circle is:

$$r^2 = (x-h)^2 + (y-k)^2$$

where

h = x coordinate of circle center
k = y coordinate of circle center
r = cicle radius,

To find the y coordinate, X from the slope intercept equation is substitued into the X from the circle equation and solved for Y.

$$r^2 = (Y/m + a - h)^2 + (Y - k)^2$$

Expanding and putting in the quadradic form

$$(1/m^2 + 1)Y^2 + 2[(a-h)/m - k]Y + (a-h)^2 + k^2 - r^2 = 0.$$

Solving this for Y

$$Y = \frac{-\left(\frac{(a-h)}{m} - k\right) + \sqrt{r^2\left(\frac{1}{m^2} + 1\right) - \left((a-h) + \frac{k}{m}\right)^2}}{\left(\frac{1}{m^2} + 1\right)}.$$

<u>X</u> <u>Coordinate</u>. Since the value for Y is determined first, the X value can now be determined from

$$X = Y/m + a.$$

<u>Incident</u> <u>Disk</u> <u>Rest</u> <u>Point</u> <u>Coordinates</u>

The incident disk rest point coordinates are used by the roll and rest point determination algorythm discussed in Section V. The formulas for these coordinates are used in the TFG Simulator and are derived here (See Appendix A for the coded routine and Figure 3.11 for a diagram).

<u>X</u> <u>Coordinate</u>. Given points P' and P" the line A passing through them can be written as:

$$\frac{Y - Y''}{X - X''} = \frac{Y' - Y''}{X' - X''}$$

However, in this case $X''=0$ and $Y''=0$. So the equation becomes

$$Y = Y'X/X'.$$

Since B is perpendicular to A

$$\text{slope of } B = -1/m.$$

Also, since $X''=0$ and $Y''=0$ the intersection of A and B is at

$$X_c = X'/2 \quad \text{and} \quad Y_c = Y'/2.$$

With this the equation for line B becomes

$$X - Y_c = -(X - X_c).$$

Substituting and reorganizing

$$Y = \frac{-X'}{Y'}X + \frac{X'^2}{2Y'} + \frac{Y'}{2}. \qquad (1)$$

or

$$X = -\frac{Y'}{X'}Y + \frac{Y'^2}{2X'} + \frac{X'}{2}. \qquad (2)$$

Since the center of C lies on B and circle C", the y coordinate for the C center can now be found by substituting the Y above into the equation for circle C".

$$C'' = Y^2 + X^2 = 2 \qquad (3)$$

$$C'' = \left(-\frac{X'}{Y'}X + \frac{X'^2}{2Y'} + \frac{Y'}{2}\right)^2 + X^2 = 2$$

Expanding and putting into quadratic form

$$X^2[4R] - X[4X'R] + [R^2 - 8Y'^2] = 0$$

where

$$R = (X'^2 + Y'^2).$$

Solving this for X using the quadratic formula

$$X = \frac{X'R \pm \sqrt{X'^2R^2 - R(R^2 - 8Y'^2)}}{2R}.$$

By working with this equation for X, it soon becomes apparent that the plus is used when X' is greater than zero.

<u>Y Coordinate</u>. The y coordinate is determined similarly by substituting X from equation 2 into equation 3. The resulting equation is

$$Y = \frac{Y'R \pm \sqrt{Y'^2R^2 - R(R^2 - 8X'^2)}}{2R}.$$

By working with this equation for Y, it soon becomes apparent that the plus is used when Y' is less than zero.

Density Definitions and Derivations

The density defintions and derivations presented here are used in the analysis portion of the TFG Simulator and

discussed in Section V. The actual coded routines are presented in Appendix A.

The density of any given computer generated material is defined as

$$DEN = DSKCNT/CELCNT$$

where

DSKCNT = number of disks
CELCNT = number of cells.

Consequently, the absolute density is defined as follows:

$$ADEN = (\text{total disk area})/(\text{total film area})$$
$$= [DSKCNT \times (\text{area/disk})]/[CELCNT \times (\text{area/cell})].$$

However,

$$\text{area/disk} = \pi R^2 = \pi(\sqrt{2}/2)^2 = \pi/2$$

and

$$\text{area/cell} = 1.$$

Therefore,

$$ADEN = (\pi/2) \times DEN = (\pi \times DSKCNT)/(2 \times CELCNT).$$

In addition, the maximum absolute density can be determined by considering the absolute density of a hexagonal close pack substance. Maximum absolute density is

$$MADEN = \frac{DSKCNT \times (\text{area/disk})}{CELCNT \times (\text{area/cell})}$$

where

$DSKCNT \times (\text{area/disk}) = 3\pi/2$
$CELCNT \times (\text{area/cell}) = \text{hexagonal area} = 3\sqrt{3}.$

Therefore, the maximum absolute density is

$$MADEN = \pi/(2\sqrt{3}).$$

Finally, the relative density is defined as

$$RDEN = ADEN/MADEN = \sqrt{3} \times DEN$$

where

$$ADEN = (\pi \times DSKCNT)/(2 \times CELCNT)$$
$$MADEN = \pi/(2\sqrt{3}).$$

Bibliography

1. Berry, Robert W., Peter M. Hall and Murray T. Harris, Thin Film Technology: 19-25. Van Nostrand Reinhold Company, New York (1968).

2. Cargill, G. S. III, "Anisotropic Microstructure in Evaporated Amorphous Germanium Films," Physical Review Letters, 28 (21): 1372-1375 (May 1972).

3. Dirks, A. G. and H. J. Leamy, "Columnar Microstructure in Vapor-Deposited Thin Films," Thin Solid Films, 47: 219-233 (1977).

4. Donovan, Terence M. and Klaus Heinemann, "High-Resolution Electron Microscope Observation of Voids in Amorphous Ge," Physical Review Letters, 27 (26): 1794-1796 (December 1971).

5. Hara, Kazuhiro, Hiroshi Fujiwara, Takashi Hashimoto, Kunito Okamoto and Takashi Hashimoto, "Crystallographic Investigations of Columnar Grains in Iron Film Evaporated at Oblique Incidence," Journal of the Physical Society of Japan, 39 (5): 1252-1255 (November 1975).

6. Hashimoto, Takashi, "Magnetic Anisotropy in Nickel Films Evaporated at Oblique Incidence," Journal of the Physical Society of Japan, 41 (2): 454-458 (August 1976).

7. Hashimoto, Takashi, Kazuhiro Hara, Kunito Okamoto and Hiroshi Fujiwara, "Top Shape of Columns in Oblique Incidence Iron Films, Journal of the Physical Society of Japan, 41 (4): 1433-1434 (October 1976).

8. Hashimoto, Takashi, Kazuhiro Hara and Eiji Tatsumoto, "Columnar Grain Structure in 76 Permalloy Films Evaporated at Oblique Incidence," Journal of the Physical Society of Japan, 24: 1400 (1968).

9. Leamy, H. J. and A. G. Dirks, "Microstructure and Magnetism in Amorphous Rare-Earth-Transition-Metal Thin Films. I. Microstructure," Journal of Applied Physics, 49 (6): 3430-3438 (June 1978).

10. Leamy, H. J., G. H. Gilmer and A. G. Dirks, "The Microstructure of Vapor Deposited Thin Films," Current Topics in Materials Science, Volume 6 (4): 311-344 (1980).

11. Macleod, H. Angus, private communication, 1984.

12. Metzdorf, W. and H. E. Wiehl, "Negative Oblique-Incidence Anisotropy in Magnetostriction-Free Permalloy Films," Phys. Stat. Sol., 17: 285-293 (1966).

13. Moss, S. C. and J. F. Graczyk, "Evidence of Voids Within the As-Deposited Structure of Glassy Silicon," Physical Review Letters, 23 (20): 1167-1170 (November 1969).

14. Nakhodkin, N. G., A. F. Bardamid and A. I. Novoselskaya, "Effects of the Angle of Deposition on Short-Range Order in Amorphous Germanium," Thin Solid Films, 112: 267-277 (1984).

15. Nakhodkin, N. G. and A. I. Shaldervan, "Effect of Vapour Incidence Angles on Profile and Properties of Condensed Films," Thin Solid Films, 10: 109-122 (1972).

16. Nakhodkin, N. G., A. I. Shaldervan, A. F. Bardamid and S. P. Chenakin, "Structural Peculiarities of Amorphous Germanium Films," Thin Solid Films, 34: 21-25 (1976).

17. Nieuwenhuizen, J. M. and H. B. Haanstra, Philips Techn. Rev., 27: 87 (1966).

18. Okamoto, Kunito, Takashi Hashimoto, Kazuhiro Hara, Hiroshi Fujiwara and Takashi Hashimoto, "Substrate Temperature Dependence Of Oblique Incidence Anisotropy in Fe Films," Journal of the Physical Society of Japan, 34: 1102-1103 (1973).

19. Okamoto, Kunito, Takashi Hashimoto, Kazuhiro Hara and Eiji Tatsumoto, "Origin of Magnetic Anisotropy of Iron Films Evaporated at Oblique Incidence," Journal of the Physical Society of Japan, 31 (5): 1374-1379 (November 1971).

20. Outlaw, R. A. and J. H. Heinbockel, "A Potential Energy Scaling Monte Carlo Simulation of Thin Film Nucleation and Growth,"Thin Solid Films, 108: 79-86 (1983).

21. Pandya, D. K., A. C. Rastogi and K. L. Chopra, "Obliquely Deposited Amorphous Ge Films. I. Growth and Structure," Journal of Applied Physics, 46 (7): 2966-2975 (July 1975).

22. Patel, S. M. and A. D. Patel, "Growth of $AgInSe_2$ Thin Films," Thin Solid Films, 111: 53-58 (1984).

23. Rudee, M. L., "Structural Variation in Amorphous Ge," Phil. Mag., 28: 1149-1151 (1973).

24. Wade, R. H. and J. Silcox, "Small Angle Electron Scattering from Vacuum Condensed Metallic Films (I)," Phys. Stat. Sol., 19: 57-62 (1967).

25. Wade, R. H. and J. Silcox, "Small Angle Electron Scattering from Vacuum Condensed Metallic Films (II)," Phys. Stat. Sol., 19: 63-76 (1967).

VITA

Captain Jeffrey A. Stefoneck was born on 25 November 1951 in Rhinelander, Wisconsin. He graduated from Rhinelander High School in 1970 and then attended the University of Wisconsin from which he received the degree of Bachelor of Science in Physics in May 1974. Upon graduation, he received a commission in the USAF through the ROTC program and was called to active duty in October 1974. He completed Missile Combat Crew Readiness training and then served four years with the 90th Strategic Missile Wing as a missile combat crew member, F.E. Warren AFB, Wyoming. While at F.E. Warren AFB, he completed a variety of courses in Business Management and Electrical Engineering from the University of Wyoming and Colorado State University respectively. In March of 1979 he attended the School of Systems and Logistics, Air Force Institute of Technology (AFIT). After completion of the Systems Acquisition Short Course he worked with the Strategic Communications Systems Program Offices, Electronic Systems Division, Hanscom AFB, Massachusetts, until entering the School of Engineering, AFIT, in May 1983.

Permanent address: 1140 N. Stevens St
Rhinelander, Wisconsin 54501

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GEP/PH/84D-10

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Engineering

6b. OFFICE SYMBOL (If applicable): AFIT/ENP

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433

7b. ADDRESS (City, State and ZIP Code):

8a. NAME OF FUNDING/SPONSORING ORGANIZATION:

8b. OFFICE SYMBOL (If applicable):

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS (City, State and ZIP Code):

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification): COMPUTER MODELING OF THIN FILM GROWTH

12. PERSONAL AUTHOR(S): Stefoneck, Jeffrey A., Capt., USAF

13a. TYPE OF REPORT: MS Thesis

13b. TIME COVERED: FROM ________ TO ________

14. DATE OF REPORT (Yr., Mo., Day): 1984 December

15. PAGE COUNT: 105

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB. GR.
20	06	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Thin Films, Thin Film Growth, Computer Simulations

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisor: John J. Wharton, Major, USAF

Approved for public release: IAW AFR 190-1?.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: John J. Wharton, Maj., USAF

22b. TELEPHONE NUMBER (Include Area Code): 513-255-4498

22c. OFFICE SYMBOL: AFIT/ENP

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

A two dimensional, hard disk computer model has been made which simulates thin film growth. The model represents deposition molecules by hard disks, which are trajected at some angle to the substrate. At the substrate, the model assumes a limited mobility where incident molecules are captured upon contact and then allowed to move to the nearest rest pocket. The model monitors disk movement by organizing the deposition field into a 320 by 240 array.

An analysis of nine different deposition angles shows that structural anisotropy and voids are a natural occurrence of the deposition process. The amount of unfilled space and the anisotropy can be linked to the deposition angle and mobility of the incident particles.

END
FILMED

6-86

DTIC